

Java Script: JQuery and Bootstrap

Sunnie Chung

Adding jQuery to Your Web Pages

There are several ways to start using jQuery on your web site. You can:

- Download the jQuery library from jQuery.com
 - Include jQuery from a CDN, like Google
-

Downloading jQuery

There are two versions of jQuery available for downloading:

- Production version - this is for your live website because it has been minified and compressed
- Development version - this is for testing and development (uncompressed and readable code)

Both versions can be downloaded from jQuery.com.

The jQuery library is a single JavaScript file, and you reference it with the HTML `<script>` tag (notice that the `<script>` tag should be inside the `<head>` section):

```
<head>  
<script src="jquery-3.1.1.min.js"></script>  
</head>
```

Tip: Place the downloaded file in the same directory as the pages where you wish to use it.

Do you wonder why we do not have `type="text/javascript"` inside the `<script>` tag?

This is not required in HTML5. JavaScript is the default scripting language in HTML5 and in all modern browsers!

jQuery CDN

If you don't want to download and host jQuery yourself, you can include it from a CDN (Content Delivery Network).

Both Google and Microsoft host jQuery.

To use jQuery from Google or Microsoft, use one of the following:

Google CDN:

```
<head>  
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>  
</head>
```

Microsoft CDN:

```
<head>  
<script src="https://ajax.aspnetcdn.com/ajax/jquery/jquery-3.1.1.min.js"></script>  
</head>
```

One big advantage of using the hosted jQuery from Google or Microsoft:

Many users already have downloaded jQuery from Google or Microsoft when visiting another site. As a result, it will be loaded from cache when they visit your site, which leads to faster loading time. Also, most CDN's will make sure that once a user requests a file from it, it will be served from the server closest to them, which also leads to faster loading time.

What is AJAX?

AJAX = Asynchronous JavaScript and XML.

In short; AJAX is about loading data in the background and display it on the webpage, without reloading the whole page.

Examples of applications using AJAX: Gmail, Google Maps, Youtube, and Facebook tabs.

You can learn more about AJAX in our [AJAX tutorial](#).

What About jQuery and AJAX?

jQuery provides several methods for AJAX functionality.

With the jQuery AJAX methods, you can request text, HTML, XML, or JSON from a remote server using both HTTP Get and HTTP Post - And you can load the external data directly into the selected HTML elements of your web page!

Without jQuery, AJAX coding can be a bit tricky!

Writing regular AJAX code can be a bit tricky, because different browsers have different syntax for AJAX implementation. This means that you will have to write extra code to test for different browsers. However, the jQuery team has taken care of this for us, so that we can write AJAX functionality with only one single line of code

jQuery - AJAX load() Method

jQuery load() Method

The jQuery load() method is a simple, but powerful AJAX method.

The load() method loads data from a server and puts the returned data into the selected element.

Syntax:

```
$(selector).load(URL,data,callback);
```

The required URL parameter specifies the URL you wish to load.

The optional data parameter specifies a set of querystring key/value pairs to send along with the request.

The optional callback parameter is the name of a function to be executed after the load() method is completed.

Here is the content of our example file: "demo_test.txt":

```
<h2>jQuery and AJAX is FUN!!!</h2>
< p id="p1">This is some text in a paragraph.</p>
```

The following example loads the content of the file "demo_test.txt" into a specific <div> element:

Example

```
$("#div1").load("demo_test.txt");
```

It is also possible to add a jQuery selector to the URL parameter.

The following example loads the content of the element with id="p1", inside the file "demo_test.txt", into a specific <div> element:

Example

```
$("#div1").load("demo_test.txt #p1");
```

The optional callback parameter specifies a callback function to run when the load() method is completed. The callback function can have different parameters:

- responseTxt - contains the resulting content if the call succeeds
- statusTxt - contains the status of the call
- xhr - contains the XMLHttpRequest object

The following example displays an alert box after the load() method completes. If the load() method has succeeded, it displays "External content loaded successfully!", and if it fails it displays an error message:

Example

```
$("#button").click(function(){
    $("#div1").load("demo_test.txt", function(responseTxt, statusTxt, xhr){
        if(statusTxt == "success")
            alert("External content loaded successfully!");
        if(statusTxt == "error")
            alert("Error: " + xhr.status + ": " + xhr.statusText);
    });
});
```

jQuery - AJAX get() and post() Methods

The jQuery get() and post() methods are used to request data from the server with an HTTP GET or POST request.

HTTP Request: GET vs. POST

Two commonly used methods for a request-response between a client and server are: GET and POST.

- **GET** - Requests data from a specified resource
- **POST** - Submits data to be processed to a specified resource

GET is basically used for just getting (retrieving) some data from the server. **Note:** The GET method may return cached data.

POST can also be used to get some data from the server. However, the POST method NEVER caches data, and is often used to send data along with the request.

To learn more about GET and POST, and the differences between the two methods, please read our [HTTP Methods GET vs POST](#) chapter.

jQuery \$.get() Method

The \$.get() method requests data from the server with an HTTP GET request.

Syntax:

```
$.get(URL,callback);
```

The required URL parameter specifies the URL you wish to request.

The optional callback parameter is the name of a function to be executed if the request succeeds.

The following example uses the \$.get() method to retrieve data from a file on the server:

Example

```
$("#button").click(function(){
    $.get("demo_test.asp", function(data, status){
        alert("Data: " + data + "\nStatus: " + status);
    });
});
```

The first parameter of \$.get() is the URL we wish to request ("demo_test.asp").

The second parameter is a callback function. The first callback parameter holds the content of the page requested, and the second callback parameter holds the status of the request.

Tip: Here is how the ASP file looks like ("demo_test.asp"):

```
<%
response.write("This is some text from an external ASP file.")
%>
```

jQuery \$.post() Method

The \$.post() method requests data from the server using an HTTP POST request.

Syntax:

```
$.post(URL,data,callback);
```

The required URL parameter specifies the URL you wish to request.

The optional data parameter specifies some data to send along with the request.

The optional callback parameter is the name of a function to be executed if the request succeeds.

The following example uses the \$.post() method to send some data along with the request:

Example

```
$("#button").click(function(){  
    $.post("demo_test_post.asp",  
    {  
        name: "Donald Duck",  
        city: "Duckburg"  
    },  
    function(data, status){  
        alert("Data: " + data + "\nStatus: " + status);  
    });  
});
```

The first parameter of \$.post() is the URL we wish to request ("demo_test_post.asp").

Then we pass in some data to send along with the request (name and city).

The ASP script in "demo_test_post.asp" reads the parameters, processes them, and returns a result.

The third parameter is a callback function. The first callback parameter holds the content of the page requested, and the second callback parameter holds the status of the request.

Tip: Here is how the ASP file looks like ("demo_test_post.asp"):

```
<%  
dim fname,city  
fname=Request.Form("name")  
city=Request.Form("city")  
Response.Write("Dear " & fname & ". ")
```

```
Response.Write("Hope you live well in " & city & ".")
%>
```

jQuery AJAX Reference

For a complete overview of all jQuery AJAX methods, please go to our [jQuery AJAX Reference](#).

What is Bootstrap?

- Bootstrap is a free front-end framework for faster and easier web development
- Bootstrap includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many other, as well as optional JavaScript plugins
- Bootstrap also gives you the ability to easily create responsive designs

What is Responsive Web Design?

Responsive web design is about creating web sites which automatically adjust themselves to look good on all devices, from small phones to large desktops.

Bootstrap History

Bootstrap was developed by Mark Otto and Jacob Thornton at Twitter, and released as an open source product in August 2011 on GitHub.

In June 2014 Bootstrap was the No.1 project on GitHub!

Why Use Bootstrap?

Advantages of Bootstrap:

- **Easy to use:** Anybody with just basic knowledge of HTML and CSS can start using Bootstrap
 - **Responsive features:** Bootstrap's responsive CSS adjusts to phones, tablets, and desktops
 - **Mobile-first approach:** In Bootstrap 3, mobile-first styles are part of the core framework
 - **Browser compatibility:** Bootstrap is compatible with all modern browsers (Chrome, Firefox, Internet Explorer, Safari, and Opera)
-

Where to Get Bootstrap?

There are two ways to start using Bootstrap on your own web site.

You can:

- Download Bootstrap from getbootstrap.com
 - Include Bootstrap from a CDN
-

Downloading Bootstrap

If you want to download and host Bootstrap yourself, go to getbootstrap.com, and follow the instructions there.

Bootstrap CDN

If you don't want to download and host Bootstrap yourself, you can include it from a CDN (Content Delivery Network).

MaxCDN provides CDN support for Bootstrap's CSS and JavaScript. You must also include jQuery:

MaxCDN:

```
<!-- Latest compiled and minified CSS -->
```

```
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
```

```
<!-- jQuery library -->
```

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
```

```
<!-- Latest compiled JavaScript -->
```

```
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
```

One advantage of using the Bootstrap CDN:

Many users already have downloaded Bootstrap from MaxCDN when visiting another site. As a result, it will be loaded from cache when they visit your site, which leads to faster loading time. Also, most CDN's will make sure that once a user requests a file from it, it will be served from the server closest to them, which also leads to faster loading time.

Example:


```
<!-- Bootstrap core JavaScript
===== -->
<!-- Placed at the end of the document so the pages load faster -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
<script src="<?php echo $live_site.'layout/js/bootstrap.min.js'; ?>"></script>
<!-- Just to make our placeholder images work. Don't actually copy the next line! -->
<script src="<?php echo $live_site.'layout/js/holder.js' ;?>"></script>
<!-- IE10 viewport hack for Surface/desktop Windows 8 bug -->
<script src="<?php echo $live_site.'layout/js/ie10-viewport-bug-workaround.js' ;?> "></script>
```

Create First Web Page With Bootstrap

1. Add the HTML5 doctype

Bootstrap uses HTML elements and CSS properties that require the HTML5 doctype.

Always include the HTML5 doctype at the beginning of the page, along with the lang attribute and the correct character set:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
  </head>
</html>
```

2. Bootstrap 3 is mobile-first

Bootstrap 3 is designed to be responsive to mobile devices. Mobile-first styles are part of the core framework.

To ensure proper rendering and touch zooming, add the following `<meta>` tag inside the `<head>` element:

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

The `width=device-width` part sets the width of the page to follow the screen-width of the device (which will vary depending on the device).

The `initial-scale=1` part sets the initial zoom level when the page is first loaded by the browser.

3. Containers

Bootstrap also requires a containing element to wrap site contents.

There are two container classes to choose from:

1. The `.container` class provides a responsive **fixed width container**
2. The `.container-fluid` class provides a **full width container**, spanning the entire width of the viewport

Note: Containers are not nestable (you cannot put a container inside another container).

Two Basic Bootstrap Pages

The following example shows the code for a basic Bootstrap page (with a responsive fixed width container):

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
</head>
<body>

<div class="container">
  <h1>My First Bootstrap Page</h1>
  <p>This is some text.</p>
</div>

</body>
</html>
```

The following example shows the code for a basic Bootstrap page (with a full width container):

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
</head>
<body>

<div class="container-fluid">
  <h1>My First Bootstrap Page</h1>
  <p>This is some text.</p>
</div>

</body>
</html>
```

Bootstrap Buttons

Button Styles

Bootstrap provides seven styles of buttons:

Default Primary Success Info Warning Danger Link

To achieve the button styles above, Bootstrap has the following classes:

- .btn-default
- .btn-primary
- .btn-success
- .btn-info
- .btn-warning
- .btn-danger
- .btn-link

The following example shows the code for the different button styles:

Example

```
<button type="button" class="btn btn-default">Default</button>
<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-link">Link</button>
```

The button classes can be used on an `<a>`, `<button>`, or `<input>` element:

Example

```
<a href="#" class="btn btn-info" role="button">Link Button</a>
<button type="button" class="btn btn-info">Button</button>
<input type="button" class="btn btn-info" value="Input Button">
<input type="submit" class="btn btn-info" value="Submit Button">
```

Why do we put a # in the href attribute of the link?

Since we do not have any page to link it to, and we do not want to get a "404" message, we put # as the link. In real life it should of course been a real URL to the "Search" page.

Bootstrap's Default Settings

Form controls automatically receive some global styling with Bootstrap:

All textual `<input>`, `<textarea>`, and `<select>` elements with class `.form-control` have a width of 100%.

Bootstrap Form Layouts

Bootstrap provides three types of form layouts:

- Vertical form (this is default)
- Horizontal form
- Inline form

Standard rules for all three form layouts:

- Wrap labels and form controls in `<div class="form-group">` (needed for optimum spacing)
 - Add class `.form-control` to all textual `<input>`, `<textarea>`, and `<select>` elements
-

Bootstrap Vertical Form (default)

Email:

Password:

Remember me

Submit

The following example creates a vertical form with two input fields, one checkbox, and a submit button:

Example

```
<form>
  <div class="form-group">
    <label for="email">Email address:</label>
    <input type="email" class="form-control" id="email">
  </div>
  <div class="form-group">
    <label for="pwd">Password:</label>
    <input type="password" class="form-control" id="pwd">
  </div>
  <div class="checkbox">
    <label><input type="checkbox"> Remember me</label>
  </div>
  <button type="submit" class="btn btn-default">Submit</button>
</form>
```

```
<input type="email" class="form-control" id="email" placeholder="Enter email" />
</div>
<div class="form-group">
  <label for="pwd">Password:</label>
  <input type="password" class="form-control" id="pwd" placeholder="Enter password" />
</div>
<div class="checkbox">
  <label><input type="checkbox"> Remember me</label>
</div>
<button type="submit" class="btn btn-default">Submit</button>
</form>
</div>
</body>
</html>
```

Vertical (basic) form

Email:

Password:

Remember me

After pushing Submit button: the default page is displayed

```
<input type="email" class="form-control" id="email" placeholder="Enter email" />
</div>
<div class="form-group">
  <label for="pwd">Password:</label>
  <input type="password" class="form-control" id="pwd" placeholder="Enter password" />
</div>
<div class="checkbox">
  <label><input type="checkbox"> Remember me</label>
</div>
<button type="submit" class="btn btn-default">Submit</button>
</form>
</div>
</body>
</html>
```

Simplify App Integration
Innovate and Scale with Ease. Choreograph Microservices and APIs. Go to tibco.com/app-integration

STOP The file you asked for does not exist

References:

http://www.w3schools.com/bootstrap/bootstrap_get_started.asp

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

http://www.w3schools.com/bootstrap/bootstrap_forms.asp

http://www.w3schools.com/bootstrap/bootstrap_forms_inputs.asp