

# \$.ajax({ async:true, url:url, success: function(v), error: function(xhr,status,err), timeout:3000});

The ajax() method is used to perform an AJAX (asynchronous HTTP) request. All jQuery AJAX methods use the ajax() method. This method is mostly used for requests where the other methods cannot be used.

- [The basic syntax of \\$.ajax\(\)](#)
- [Example 1: Load a text and replace in div1](#)
- [Example2: alert \( success / error\)](#)
- [Example3: Set Post result](#)
- [Example 4: Load URL](#)

## The basic syntax of \$.ajax()

```
$.ajax({name:value, name:value, ... })
```

Possible names/values in the table below:

Name	Value/Description
async	A Boolean value indicating whether the request should be handled asynchronous or not. Default is true
beforeSend(xhr)	A function to run before the request is sent
cache	A Boolean value indicating whether the browser should cache the requested pages. Default is true
complete(xhr,status)	A function to run when the request is finished (after success and error functions)
contentType	The content type used when sending data to the server. Default is: "application/x-www-form-urlencoded"
context	Specifies the "this" value for all AJAX related callback functions
data	Specifies data to be sent to the server
dataFilter(data,type)	A function used to handle the raw response data of the XMLHttpRequest
dataType	The data type expected of the server response.
error(xhr,status,error)	A function to run if the request fails.
global	A Boolean value specifying whether or not to trigger global AJAX event handles for the request. Default is true
ifModified	A Boolean value specifying whether a request is only successful if the response has changed since the last request. Default is: false.
jsonp	A string overriding the callback function in a jsonp request
jsonpCallback	Specifies a name for the callback function in a jsonp request
password	Specifies a password to be used in an HTTP access authentication request.
processData	A Boolean value specifying whether or not data sent with the request should be transformed into a query string. Default is true
scriptCharset	Specifies the charset for the request
success(result,status,xhr)	A function to be run when the request succeeds
timeout	The local timeout (in milliseconds) for the request
traditional	A Boolean value specifying whether or not to use the traditional style of param serialization
type	Specifies the type of request. (GET or POST)
url	Specifies the URL to send the request to. Default is the current page
username	Specifies a username to be used in an HTTP access authentication request
xhr	A function used for creating the XMLHttpRequest object

## Example 1: Load a text and replace in div1

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<div id="test"></div>
<script>
$( "button" ).click(function(){
    $.ajax({url: "demo_test.txt", success: function(result){
        $("#test").html(result);
    }});
});
</script>
```

## Example2: alert ( success / error)

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
// Assign handlers immediately after making the request,
// and remember the jqXHR object for this request
var jqxhr = $.ajax( "example.php" )
    .done(function() {
        alert( "success" );
    })
    .fail(function() {
        alert( "error" );
    })
    .always(function() {
        alert( "complete" );
    });

// Perform other work here ...

// Set another completion function for the request above
jqxhr.always(function() {
    alert( "second complete" );
});
</script>
```

## Example3: Set Post result

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$.ajax({
    method: "POST",
    url: "some.php",
    data: { name: "John", location: "Boston" }
})
    .done(function( msg ) {
        alert( "Data Saved: " + msg );
    });
</script>
```

## Example 4: Load URL

```

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<div id="vmap1"></div>
<script>
let url = "https://ckii.com";
let vmap_obj = "#vmap1";
let msg = get_status_msg( "black", "orange", "no creatives available (zero ad_breaks received)");

$.ajax({
  async: true,
  url: url,
  success: function(data)
  {
    // console.log(url);
    // console.log(data);
    // console.log(data.childNodes.length);
    if (data.childNodes.length>0)
    {
      // console.log("nodeName[0] = " + data.childNodes[0].nodeName);
      // console.log(data.childNodes[0].childNodes.length);
      if (data.childNodes[0].childNodes.length>0)
      {
        let ad_break_cnt=0;
        for(var i=0; i<data.childNodes[0].childNodes.length; i++)
        {
          if (data.childNodes[0].childNodes[0].nodeName.match(/AdBreak/))
            ad_break_cnt++;
        }
        if (ad_break_cnt>0)
        {
          msg = get_status_msg( "black", "#00ff00", ad_break_cnt + " ad_breaks
received");
        }
      }
    }
    else
    {
      msg = get_status_msg( "#ffffff", "green", "no vmap childnode");
    }
    $(vmap_obj).html( msg );
    // console.log( msg);
  },
  error: function(xhr,status,error)
  {
    msg = get_status_msg( "#ffffff", "red", "server error: " + status);
    $(vmap_obj).html( msg );
    // console.log( msg);
  }
});
</script>

```