$(window).unload wait for AJAX call to finish before leaving a webpage [duplicate]

Asked 14 years, 5 months ago Modified 4 years, 3 months ago Viewed 44k times

33

This question already has answers here:

JavaScript, browsers, window close - send an AJAX request or run a script on window closing (11 answers)

Closed 7 years ago.

Basically, once a user leaves a webpage in my application, I need to call a PHP script with AJAX, which will insert a time spent on the webpage to the database and then leave the page.

It is important to wait for the AJAX request to finish because webpages in my application are not accessible to users unless they have spent a certain time on a previous page (let's say two minutes).

Here is my jquery code:

```
$(document).ready(function() {

    var teid = TEID;
    var startTime = new Date().getTime();

    $(window).unload(function() {
        var timeSpentMilliseconds = new Date().getTime() - startTime;
        var t = timeSpentMilliseconds / 1000 / 60;

        $.ajax({
            type: 'POST',
            url: '/clientarea/utils/record-time',
            data: 'teid=' + teid + '&t=' + t
        });
    });

});
```

How should I change it so it will wait for the AJAX request to end before leaving the webpage?

EDIT:

Or it might be better (easier) to just let the AJAX request be repeated every minute or so. Is that possible?

javascript jquery ajax

Share
Improve this question
Follow

edited Jun 29, 2010 at 7:22

asked Jun 29, 2010 at 6:59

Richard Knop
83.6k ● 153 ● 398 ● 558

8 Answers

Sorted by: Highest score (default)

32

Well, you can set `async: false` on your AJAX call to make the browser wait for the request to finish before doing anything else, but note that this will 'hang' the browser for the duration of the request.

```
$.ajax({
    type: 'POST',
    async: false,
    url: '/clientarea/utils/record-time',
    data: 'teid=' + teid + '&t=' + t
});
```

From the manual:

> By default, all requests are sent asynchronous (i.e. this is set to true by default). If you need synchronous requests, set this option to false. Cross-domain requests and dataType: "jsonp" requests do not support synchronous operation. Note that synchronous requests may temporarily lock the browser, disabling any actions while the request is active.

⚠ WARNING: This answer was posted in 2010 and is now outdated - the XHR specification highlights the following statement:

> **Synchronous XMLHttpRequest outside of workers is in the process of being removed from the web platform** as it has detrimental effects to the end user's experience. (This is a long process that takes many years.) **Developers must not pass false for the async argument when current global object is a Window object.** User agents are strongly encouraged to warn about such usage in developer tools and may experiment with throwing an "InvalidAccessError" DOMException when it occurs.

DevTools in Chrome has recently started warning about it, so this change (which has been coming for some years) could be imminent.

Share

Improve this answer

Follow

edited Sep 14, 2020 at 6:40

mindplay.dk

7,290 ● 3 ● 48 ● 58

answered Jun 29, 2010 at 7:03

Tatu Ulmanen

125k ● 34 ● 189 ● 185

Hmm. Thanks. I have updated my question. It might be easier to just let the AJX request repeat every let's say minute. How would I do that? – Richard Knop Jun 29, 2010 at 7:06

12 I'd strongly recommend to avoid that, **especialy** on `unload`. – jAndy Jun 29, 2010 at 7:06

I have found there is always a better solution than doing `async: false`, regardless of the problem. – Jonathan M Feb 11, 2015 at 22:31

One problem I am seeing with this solution is that the request stays alive in the new page the user navigated to. – user1852503 May 22, 2015 at 13:34

11

The best solution is to use navigator.sendBeacon. It is brand new functionality which is starting to get implemented in new releases of browsers. The function is available in browsers newer than Chrome 39 and Firefox 31. It is not supported by Internet Explorer and Safari at the time of writing. To make sure your request gets send in the browsers that don't support the new functionality yet, you can use this solution:

```
var navigator.sendBeacon = navigator.sendBeacon || function (url, data) {
  var client = new XMLHttpRequest();
  client.open("POST", url, false); // third parameter indicates sync xhr
  client.setRequestHeader("Content-Type", "text/plain;charset=UTF-8");
  client.send(data);
};
```

Hope this helps!

Share Improve this answer Follow

answered May 19, 2015 at 12:55

lswartsenburg

383 ● 3 ● 8

I think this is elegant, i used it in Electron to send data when the app is closed – furiel Jan 13, 2017 at 17:44

In 2018, this is the way to go. – Baishu Feb 24, 2018 at 8:27

8

How about setting a cookie in the unload handler? The server should see it on the subsequent requests.

```
<script>
    $(window).unload(function(){document.cookie='left_on='+(new Date())})
</script>
```

Share Improve this answer Follow

answered Feb 2, 2011 at 15:12

Sixtease

790 ● 8 ● 18

This is the best solution. • The user does not have to wait for the Ajax request. • Data will be included on the next request to the server (one request, instead of two). • Less code. You can set the cookie to expire after a few seconds, and unset the cookie as soon as the page loads to avoid privacy issues with GDPR. – Frank Forte Oct 24, 2019 at 15:15

4

for me, yours is not a good idea for the browser to wait before closing... simply because what if I really want to close it?...

if a page bother a user, it's not good...

my suggestion is, in the page, you wait for 2 minutes (if 2 minutes is the requirements), then send an ajax that the user has done his 2 minutes...

you can then check it on the server side if one has his 2 minutes or not...

Share Improve this answer Follow

answered Jun 29, 2010 at 7:06

Reigel Gallarde

65.2k ● 21 ● 124 ● 141

2

It is a bad idea to try and hijack your users' browser, since it will give them a bad feeling and send them away.

If for some reason you want not to produce a new page until the user has spent a minimum time on the previous one, the best thing to do is to pilot server side, i.e. redirecting to the current page until the requested time has passed.

You don't even need to make ajax calls, just store in the session the timestamp of when the page was served, and don't show the following page until a certain amount of time has passed.

Be sure to tell the users they have to wait for a new page to be ready, maybe with a simple javascript countdown.

If you want the user to actually have the page active for a certain amount of time (i.e. not to switch to another tab/window waiting for the two minutes to elapse), well, I cannot propose an effective solution.

Share Improve this answer Follow

answered Jun 29, 2010 at 7:40

Iacopo
4,282 ● 1 ● 24 ● 25

1

use `onbeforeunload`:

```
$(document).ready(function(){
  window.onbeforeunload = function(){
     // $.ajax stuff here
     return false;
  }
});
```

This will at least bring the user a `messagebox` which asks him if he wants to close the current window/tab.

Share Improve this answer Follow

answered Jun 29, 2010 at 7:08

jAndy
236k ● 57 ● 311 ● 363

1

I think it would be much better to use a polling technique as you suggest, though it will cause some load on the web server.

```
$(document).ready(function() {

    var teid = TEID;
    var startTime = new Date().getTime();

    var ajaxFunc = function() {
        var timeSpentMilliseconds = new Date().getTime() - startTime;
        var t = timeSpentMilliseconds / 1000 / 60;

        $.ajax({
            type: 'POST',
            url: '/clientarea/utils/record-time',
            data: 'teid=' + teid + '&t=' + t
        });
    };

    setInterval(ajaxFunc, 60000);

})
```

You'll be glad when you can use websockets :)

Share Improve this answer Follow

answered Jun 29, 2010 at 7:14

sje397
41.8k ● 9 ● 89 ● 105

Thanks. But this will only get executed once after one minute? Could it execute repeatedly every minute (60 seconds)? – Richard Knop Jun 29, 2010 at 7:33

2 That's what setInterval does, as opposed to setTimeout – sje397 Jun 29, 2010 at 12:44

0

The `jQuery.ajax()` method has the option `async`. If you set it to `false` the call will block until the response comes back (or it timed out). I'm pretty shure, that calling this, will yause the browser to weit in the unload handler.

Side note: You can't rely on this to work. If the browser gives the user the option to cancel the unload handlers (which some browsers do after a while of waiting), the "time spend on site" will never be updated. You could add a timer to the site, which periodically calls a script on the server and which updates the time. You won't have an accurate value, but in your case, this isn't needed.

If you only need to know if the user was X seconds on the page You could simply set a timeout in the onload handler (using `setTimeout(function, ms)`) which makes a call if the user has spend the needed time. So there would be no need for a unload handler.

Share Improve this answer Follow

answered Jun 29, 2010 at 7:07

Martin Thurau

7,644 ● 7 ● 45 ● 80