Create and save a file with JavaScript [duplicate]

Asked 12 years, 4 months ago Modified 2 years, 11 months ago Viewed 1.1m times

356

This question already has answers here:

How to create a file in memory for user to download, but not through server? (23 answers)

Closed 4 years ago.

I have data that I want to write to a file, and open a file dialog for the user to choose where to save the file. It would be great if it worked in all browsers, but it has to work in Chrome. I want to do this all client-side.

Basically I want to know what to put in this function:

```
saveFile: function(data)
{
}
```

Where the function takes in data, has the user select a location to save the file, and creates a file in that location with that data.

Using HTML is fine too, if that helps.

javascript file dialog save

Share Improve this question Follow

edited Mar 31, 2022 at 12:45
isherwood
61.1k 16 121 169

asked Nov 15, 2012 at 19:56
user1756980
4,007 4 18 13

7 Has been an exact duplicate of Create a file in memory for user to download, not through server for years. – Dan Dascalescu Nov 7, 2016 at 20:48

9 Answers

Sorted by: Highest score (default)

348

A very minor improvement of the code by Awesomeness01 (no need for anchor tag) with addition as suggested by trueimage (support for IE):

```
// Function to download data to a file
function download(data, filename, type) {
    var file = new Blob([data], {type: type});
    if (window.navigator.msSaveOrOpenBlob) // IE10+
        window.navigator.msSaveOrOpenBlob(file, filename);
    else { // Others
        var a = document.createElement("a"),
                url = URL.createObjectURL(file);
        a.href = url;
        a.download = filename;
        document.body.appendChild(a);
        a.click();
        setTimeout(function() {
            document.body.removeChild(a);
            window.URL.revokeObjectURL(url);
        }, 0);
    }
}
```

Tested to be working properly in Chrome, FireFox and IE10.

In Safari, the data gets opened in a new tab and one would have to manually save this file.

Share Improve this answer Follow

edited May 23, 2017 at 10:31
Community Bot
1 1

answered Jun 14, 2015 at 17:09
Kanchu
3,809 1 16 16

1 Specifically which version of IE? – Aaron Yarborough Feb 8, 2016 at 13:38

@Soren It currently does not work in any version of IE, as far as I know. Safari might not have it either, but I don't work with those and I would have to look it up. – aidanjacobson Mar 31, 2016 at 3:45

Not working in Google Chrome. I am thinking to make this as a bookmarklet `javascript:function download(text, name, type) {   var a = document.getElementById("content");   var file = new Blob([text], {type: type});   a.href = URL.createObjectURL(file);`

`a.download = name; }` – user442096 Apr 1, 2016 at 10:03

1 This is not working in IE 11.0.9600.18426, but it is working in Chrome 52.0.2743.116 m. Any ideas to get this working in IE 11? – trueimage Aug 24, 2016 at 19:22

To get it working in IE 11 and Chrome I replaced the a.href line with: if(window.navigator.msSaveOrOpenBlob) { window.navigator.msSaveOrOpenBlob(file, name); } else { a.href = URL.createObjectURL(file); } – trueimage Aug 24, 2016 at 19:28

1 Seems to me like `var a = document.createElement("a")` should be in the `else` branch, otherwise IE10+ will create but not remove the anchor (`<a>`) element. – Andrew Willems Jan 22, 2017 at 21:12

This solution should be really accepted. I have created a repo for anyone who wants to try this solution. github.com/slavede/ajax-js-download It uses mentioned logic. I've tested in latest Chrome, FF, Edge and Safari. – Slaven Tomac Sep 14, 2018 at 12:49

1 Seems not to work in Safari, just nothing happens (no save) – Jonny Dec 1, 2018 at 19:31

1 Just to make your code a little neater, do `var file = new Blob([data], {type});` – Geza Kerecsenyi Apr 16, 2019 at 18:32

2 How to set location in this script when download? – Bravo Yeung Jun 4, 2019 at 4:33

1 It doesn't work in Chrome for Mobile. – Jp_ Dec 12, 2019 at 15:46

Can we specify which folder to download to? – codertryer Oct 25, 2020 at 18:44

3 What can we use for `type` ? – posfan12 Dec 26, 2020 at 3:59

This is a great solution. I tried the code and find it still works without `document.body.appendChild(a)` and `setTimeout()` . The final code is `var a = document.createElement("a"), url = URL.createObjectURL(file); a.href = url;  a.download = filename; a.click(); URL.revokeObjectURL(url);` Are there any drawbacks? – Evi Song Nov 11, 2021 at 7:31

162

```
function download(text, name, type) {
  var a = document.getElementById("a");
  var file = new Blob([text], {type: type});
  a.href = URL.createObjectURL(file);
  a.download = name;
}



<a href="" id="a">click here to download your file</a>
<button onclick="download('file text', 'myfilename.txt', 'text/plain')">Create
```

```
file</button>
```

Run code snippet Expand snippet

And you would then download the file by putting the download attribute on the anchor tag.

The reason I like this better than creating a data url is that you don't have to make a big long url, you can just generate a temporary url.

Share Improve this answer Follow

edited Apr 11, 2015 at 22:57
Community Bot
1 1

answered Mar 30, 2015 at 5:52
aidanjacobson
2,401 1 20 17

@Banjocat You should check if the browser support certain objects. Example detecting: `if("URL"in window&&"createObjectURL"in URL&&"download"in Element.prototype)` : else then you just change the downloading method or notice that the browser doesn't support the needed objects to download the file. – user5066707 Jan 20, 2016 at 22:24

1 In firefox this works if you click the link, but if you right click and choose Save Link As... nothing happens. – aamarks Apr 28, 2018 at 5:04

1 Doesn't work in Google Chrome: `Download is disallowed. The frame initiating or instantiating the download is sandboxed, but the flag ‘allow-downloads’ is not set. See https://www.chromestatus.com/feature/5706745674465280 for more details.` – posfan12 Dec 26, 2020 at 3:49

@posfan12 Try implementing it in your own page, SO renders the code snippet in an iframe, which is not allowed to download files without the allow-downloads flag. – aidanjacobson Dec 28, 2020 at 1:54

Thanks for helping out, this works for my requirement. Additionally, I need a timestamp on the downloaded document. It would be grateful if you can update on how can we add a timestamp to the file to know when it is downloaded. – User Aug 7, 2024 at 10:36

@User you can get a timestamp using Date.now() or other date methods. Then it's up to you to put it in either name or text. – aidanjacobson Aug 8, 2024 at 15:56

44

Choosing the location to save the file before creating it is not possible. But it is possible, at least in Chrome, to generate files using just JavaScript. Here is an old example of mine of creating a CSV file. The user will be prompted to download it. This, unfortunately, does not work well in other browsers, especially IE.

```
<!DOCTYPE html>
<html>
<head>
    <title>JS CSV</title>
</head>
<body>
    <button id="b">export to CSV</button>
    <script type="text/javascript">
        function exportToCsv() {
            var myCsv = "Col1,Col2,Col3\nval1,val2,val3";

            window.open('data:text/csv;charset=utf-8,' + escape(myCsv));
        }

        var button = document.getElementById('b');
        button.addEventListener('click', exportToCsv);
    </script>
</body>
</html>
```

Share Improve this answer Follow

answered Nov 15, 2012 at 20:08

Matt Greer
62.1k 18 126 124

1 When I use this it opens a new tab with the text in it, it doesn't open a file dialog window. – user1756980 Nov 15, 2012 at 20:13

@user1756980 - Yes. You would need to "Save To File" from that new tab. – Jesse Chisholm Jul 22, 2014 at 17:44

It depends on the browser, os, etc. At the time I wrote the answer, a csv data url in Chrome would pop a save dialog – Matt Greer Jul 22, 2014 at 20:53

1 @JesseChisholm you can do that in javascript. just create an anchor tag in a variable and put the download attribute on it (like so: a.download = "downloadname.txt") and then clicking it with a.click(). – aidanjacobson Mar 30, 2015 at 5:50

17

For latest browser, like Chrome, you can use the File API as in this tutorial:

```
window.requestFileSystem  = window.requestFileSystem ||
window.webkitRequestFileSystem;
window.requestFileSystem(window.PERSISTENT, 5*1024*1024 /*5MB*/, saveFile,
errorHandler);
```

Share Improve this answer Follow

answered Nov 15, 2012 at 20:05

pdjota
3,243 3 25 33

2 I think this code snippet would be closer to the asker's intention. – Fabrício Matté Nov 15, 2012 at 20:08

2 See here for evidence of it's deadness: lists.w3.org/Archives/Public/public-webapps/2014AprJun/... – voidstate Sep 30, 2014 at 13:43

14

```
function SaveBlobAs(blob, file_name) {
    if (typeof navigator.msSaveBlob == "function")
        return navigator.msSaveBlob(blob, file_name);

    var saver = document.createElementNS("http://www.w3.org/1999/xhtml", "a");
    var blobURL = saver.href = URL.createObjectURL(blob),
        body = document.body;

    saver.download = file_name;

    body.appendChild(saver);
    saver.dispatchEvent(new MouseEvent("click"));
    body.removeChild(saver);
    URL.revokeObjectURL(blobURL);
}
```

Share Improve this answer Follow

edited Jan 16, 2020 at 7:23

Hafez Divandari
9,069 5 49 65

answered Oct 12, 2019 at 16:37

user12066722
327 3 3

10

Tried this in the console, and it works.

```
var aFileParts = ['<a id="a"><b id="b">hey!</b></a>'];
var oMyBlob = new Blob(aFileParts, {type : 'text/html'}); // the blob
window.open(URL.createObjectURL(oMyBlob));
```

Share Improve this answer Follow

edited Jun 1, 2015 at 10:24

Jahid
22.4k 10 95 111

answered Apr 11, 2015 at 9:47

Netsi1964
3,620 1 29 17

4

You cannot do this purely in Javascript. Javascript running on browsers does not have enough permission yet (there have been proposals) due to security reasons.

Instead, I would recommend using Downloadify:

> A tiny javascript + Flash library that enables the creation and download of text files without server interaction.

You can see a simple demo here where you supply the content and can test out saving/cancelling/error handling functionality.

Share Improve this answer Follow

edited Dec 9, 2013 at 2:22

answered Nov 15, 2012 at 20:02

Aamir
5,440 2 31 47

Obsolete answer – Joshua Oct 6, 2023 at 20:09

4

StreamSaver is an alternative to save very large files without having to keep all data in the memory.
In fact it emulates everything the server dose when saving a file but all client side with service worker.

You can either get the writer and manually write Uint8Array's to it or pipe a binary readableStream to the writable stream

There is a few example showcasing:

- How to save multiple files as a zip
- piping a readableStream from eg `Response` or `blob.stream()` to StreamSaver
- manually writing to the writable stream as you type something
- or recoding a video/audio

Here is an example in it's simplest form:

```
const fileStream = streamSaver.createWriteStream('filename.txt')

new Response('StreamSaver is awesome').body
  .pipeTo(fileStream)
  .then(success, error)
```

If you want to save a blob you would just convert that to a readableStream

```
new Response(blob).body.pipeTo(...) // response hack
blob.stream().pipeTo(...) // feature reference
```

Share Improve this answer Follow

answered May 31, 2019 at 19:41
Endless
38k 13 116 137

For Chrome and Firefox, I have been using a purely JavaScript method.

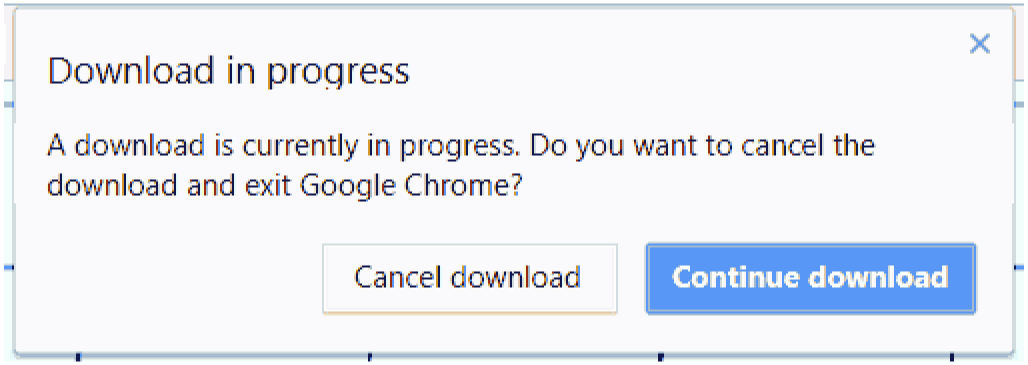
3

(My application cannot make use of a package such as `Blob.js` because it is served from a special engine: a DSP with a WWWeb server crammed in and little room for anything at all.)

```
function FileSave(sourceText, fileIdentity) {
    var workElement = document.createElement("a");
    if ('download' in workElement) {
        workElement.href = "data:" + 'text/plain' + "charset=utf-8," +
escape(sourceText);
        workElement.setAttribute("download", fileIdentity);
        document.body.appendChild(workElement);
        var eventMouse = document.createEvent("MouseEvents");
        eventMouse.initMouseEvent("click", true, false, window, 0, 0, 0, 0, 0,
false, false, false, false, 0, null);
        workElement.dispatchEvent(eventMouse);
        document.body.removeChild(workElement);
    } else throw 'File saving not supported for this browser';
}
```

Notes, caveats, and weasel-words:

- I have had success with this code in both Chrome and Firefox clients running in Linux (Maipo) and Windows (7 and 10) environments.
- However, if `sourceText` is larger than a MB, Chrome sometimes (only sometimes) gets stuck in its own download without any failure indication; Firefox, so far, has not exhibited this behavior. The cause might be some blob limitation in Chrome. Frankly, I just don't know; if anybody has any ideas how to correct (or at least detect), please post. If the download anomaly occurs, when the Chrome browser is closed, it generates a diagnostic such as



Download in progress

A download is currently in progress. Do you want to cancel the download and exit Google Chrome?

- This code is not compatible with Edge or Internet Explorer; I have not tried Opera or Safari.

Share Improve this answer Follow

answered Feb 22, 2018 at 1:28

We B Martians
375 2 13

Start asking to get answers

Find the answer to your question by asking.

Ask question

Explore related questions

javascript file dialog save

See similar questions with these tags.