FileReader vs. window.URL.createObjectURL

Asked 5 years, 7 months ago Active 3 years, 9 months ago Viewed 27k times

55

35

I'm building a mobile website and I'd like to use the Camera API to take photos. The images should be displayed on the website and uploaded to a server. According to the introduction to the Camera API on MDN the images can be accessed and displayed on the website using `FileReader` or `window.URL.createObjectURL` . I tested these possible solutions successfully with an iPad (Safari and Chrome) and an Android tablet (Chrome and Firefox).

What is the difference between `FileReader` and `window.URL.createObjectURL` ? I think `window.URL.createObjectURL` is newer but not a standard yet. Is there a difference in the performance?

javascript file-upload

Share Improve this question Follow

asked Jul 31 '15 at 8:56

nightlyop

6,599 5 25 35

1 Answer

Active Oldest Votes

101

There is difference.

1) time

- `createObjectURL` is synchronously executed (immediately)
- `FileReader.readAsDataURL` is asynchronously executed (after some time)

2) memory usage

- `createObjectURL` returns url with hash, and store object in memory until document triggers unload event (e.g. document close) or execute `revokeObjectURL`
- `FileReader.readAsDataURL` returns `base64` that contains many characters, and use more memory than blob url, but removes from memory when you don't use it (by garbage collector)

3) support

- `createObjectURL` from IE 10 and all modern browsers
- `FileReader.readAsDataURL` from IE 10 and all modern browsers

From me, is better to use blob url's (via `createObjectURL`), it is more efficient and faster, but if you use many object urls, you need to release these urls by `revokeObjectURL` (to free memory). *For example, you can call URL.revokeObjectURL inside an Image onload handler and the Image object will keep the image data, without losing it, Nahuel Greco (c).*

Share Improve this answer Follow

edited Jun 6 '17 at 5:35

answered Jul 31 '15 at 10:11

Alex Nikulin
5,940 4 30 31

Well URL releasing isn't strictly necessery if you're not dealing with a long lived SPA, as these objects are released from memory on `document.unload`. So you may be sloppy. – Robert Koritnik Oct 19 '15 at 17:02

3 I have experience in resize of a lot of images in js, so i am accustomed to be rigorous with memory managment. – Alex Nikulin Oct 20 '15 at 7:40

2 Always a good habit, yes. I just wanted to point out information that you didn't write in your otherwise great answer. Maybe just one correction. When you say *createObjectURL is in time executed* you likely mean *createObjectURL is in synchronously executed*. It should also be worded this way as *time executed* is ambiguous. Hence *works with callback* is an everyday scenario of *asynchronous* execution. – Robert Koritnik Oct 20 '15 at 8:31

3 Note, as shown in link, you can call `URL.revokeObjectURL` inside an Image `onload` handler and the Image object will keep the image data, without losing it. Then you can manipulate the Image object without taking special care of the data, it will be handled by the usual GC. Also take in account `URL.createObjectURL` is synchronous but it seems to complete almost instantaneously. – Nahuel Greco May 23 '17 at 19:46