The magic of createObjectURL()

PRODUCT & TECH BLOG / SOFTWARE ENGINEERING

The web platform is maturing faster and faster, and we're seeing the work normally done by native desktop applications now often shifting towards web-based applications instead. Features that may appear to have little importance can be really powerful when combined together.

A good example of this is `URL.createObjectURL()`. On it's own, it really doesn't do much. Paired with the HTML5 video and audio element, or even the good old image element, it gets to be really powerful.

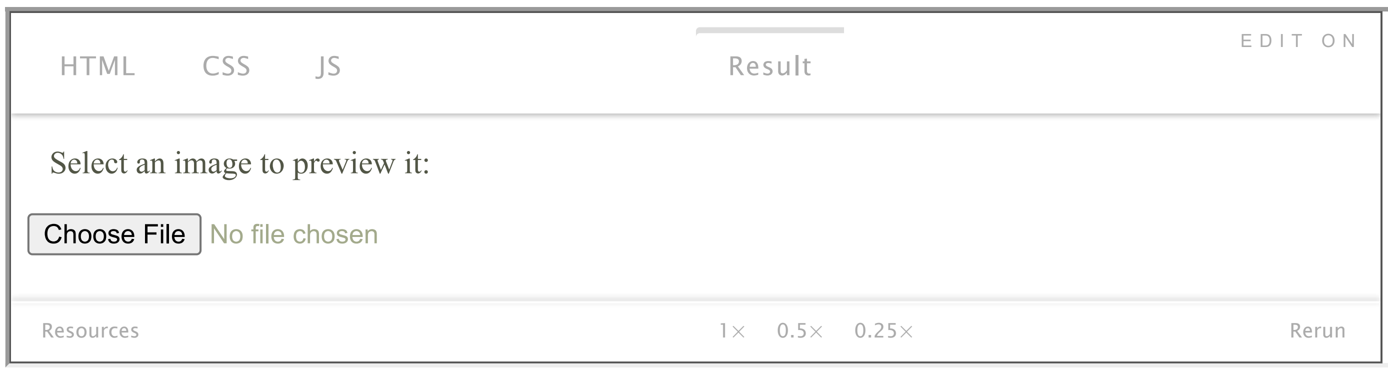
`URL.createObjectURL()` is part of the **URL-interface**, which can be used to construct and parse URLs. `URL.createObjectURL()` specifically, can be used to create a reference to a `File` or a `Blob`. As opposed to a base64-encoded data URL, it doesn't contain the actual data of the object – instead it holds a reference.

Let's say you have a page that prompts the user to upload an image. The file input element is, as we all know, very simple and far from user friendly. When a user has selected an image, it would make sense (in a lot of cases) to show a preview of the image. We are now finally able to solve this problem in a simple and intuitive way:

```
[code language="js"]
var fileInput = document.querySelector('input[type="file"]');
var preview = document.getElementById('preview');

fileInput.addEventListener('change', function(e) {
    var url = URL.createObjectURL(e.target.files[0]);
    preview.setAttribute('src', url);
});
[/code]
```

Simple, right? This produces something like the following:



Select an image to preview it:

The nice thing about this is that it's really fast. Previously, we've had to instantiate a `FileReader` instance and read the whole file as a base64 data URL, which takes time and a lot of memory. With `createObjectURL()`, the result is available straight away, allowing us to do things like reading image data to a canvas or the length of an audio file. Great, right?

One thing to keep in mind is that once you're done with the generated object URL, you should revoke it. This frees up memory, which is usually handled automatically when you close the page or navigate away from it, but it's generally a good idea anyway. Revoking the URL is a simple process – just call `URL.revokeObjectURL(<objectUrl>)` from the `onLoad`-handler of the element you are using it for.

Read more about createObjectUrl on the **Mozilla Developer Network**.

9 Comments **Schibsted** **Disqus' Privacy Policy** 1 **Login**

Recommend Tweet Share **Sort by Best**

Join the discussion…

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name

csandreas1 • 2 years ago
Can you make an example with blob image ?
1 • Reply • Share ›

Frank • 4 years ago
In reference to the above. I also noticed that the preview.onload function is not called on mobile, and on mobile it is "blob:file:///..." and local it is "blob:http://localhost/..." finally on the server it is "blob:http://www.server.com/..."
1 • Reply • Share ›

Alex Alan Nunes • a year ago
I made an example of React
Exemple
• Reply • Share ›

Carl-Erik Kopseng • 3 years ago
One trouble of directly previewing the image like this is rotation. AFAIK you do not get free EXIF rotation as part of the deal, meaning you still need to use a FileReader to extract the image, use something like exif.js to read the Exif values and finally use that to do css

image, use something like exif-js to read the Exif values and finally use that to do css transforms on the preview image.

Reply • Share ›

Tamir Shina • 3 years ago

Cool feature! and very nicely articulated.

I'm fairly new to web design and searched 'why use createobjecturl' to get to this page.

Since my base knowledge is lacking I was wondering if you can refer me to more material so I can better understand the less modern technologies and the difference to 'createobjecturl()'.

Reply • Share ›

Priyanka • 4 years ago

Thank you ,Clear Explanation

Reply • Share ›

Frank • 4 years ago

This works great on my local machine, a PC, but it does not on my mobile app created with PhoneGap. Do you if there are setting I need to adjust. The url variable I get looks good. It is blob:file:///497c4.....
There is no error but also no image displayed. I just copied your code.
I would appreciate any suggestions.
Thanks,
Frank.

Reply • Share ›

Torkel Bjørnson-Langen • 6 years ago

Sweet! Thank you for the tip – I will probably use this my self.

Just one note: If the input[type=file] allows multiple selections then this code will only preview the first. Is it possible to rewrite the querySelector() query to omit elements with the 'multiple' attribute? Or do I have to do it using xpath or javascript? Thanks!

Reply • Share ›

Espen Hovlandsdal ➦ Torkel Bjørnson-Langen • 6 years ago

Sure. This was a very contrived example. You could either exclude inputs with the multiple attribute:

```
document.querySelector('input[type="file"]:not([multiple])');
```

Or, even better, you could generate images on the fly for all images selected. See this JSFiddle for a better example.

Reply • Share ›

Subscribe   Add Disqus to your siteAdd DisqusAdd   Do Not Sell My Data

WRITTEN BY
Espen Hovlandsdal

PUBLISHED
2014-12-01

Read also

SOFTWARE ENGINEERING

How we protected ourselves from the Dependency Confusion attack

PRODUCT / SOFTWARE ENGINEERING

They make sure Sweden’s favorite news product runs smoothly

PRODUCT / SOFTWARE ENGINEERING

The Omni Next team built a platform used by four brands