

bfcache and other traps in hellish histories

Back/Forward button

inconsistencies

(just another joy of browsers)

Some examples

Classic problem #1

```
<script>
  window.onload = function() { document.location = 'another_page.html' }
</script>
```

What happens when the user goes 'back'?

- Chrome and Safari will go back to the page containing the redirect script

What happens when the user goes 'back'?

- Chrome and Safari will go back to the page containing the redirect script
- Firefox and IE will go back to the page BEFORE the page containing the redirect script (if there is one)

$$\left(\begin{smallmatrix} \text{---} & \circ\Box\circ \\ \text{---} & \end{smallmatrix}\right) \sim \left(\begin{smallmatrix} \perp & \perp \\ \perp & \end{smallmatrix}\right)$$

$$\left(\begin{smallmatrix} \text{---} & \circ\Box\circ \\ \text{---} & \end{smallmatrix}\right) \sim \left(\begin{smallmatrix} \perp & \perp \\ \perp & \end{smallmatrix}\right)$$

$$\left(\begin{smallmatrix} \text{---} & \circ\Box\circ \\ \text{---} & \end{smallmatrix}\right) \sim \left(\begin{smallmatrix} \perp & \perp \\ \perp & \end{smallmatrix}\right)$$

$$\left(\begin{smallmatrix} \text{---} & \circ\Box\circ \\ \text{---} & \end{smallmatrix}\right) \sim \left(\begin{smallmatrix} \perp & \perp \\ \perp & \end{smallmatrix}\right)$$

Classic problem #2

```
var clickCount = 0;  
$( 'button' ).on( 'click', function() {  
    clickCount += 1;  
    if (clickCount === 3) {  
        doSomethingAwesome();  
    }  
} );
```

What happens when the user goes 'back'?

- Chrome, Safari and IE will load the page from the cache, and then re-execute the javascript

What happens when the user goes 'back'?

- Chrome, Safari and IE will load the page from the cache, and then re-execute the javascript
- Firefox will load the page from the cache, but not re-execute the javascript. It retains the previous JavaScript STATE instead.

Classic problem #2

```
var clickCount = 0;  
$( 'button' ).on( 'click', function() {  
    clickCount += 1;  
    if (clickCount === 3) {  
        doSomethingAwesome();  
    }  
} );
```

$$\left(\begin{smallmatrix} \text{---} & \circ\Box\circ \\ \text{---} & \end{smallmatrix}\right) \sim \left(\begin{smallmatrix} \perp & \perp \\ \perp & \end{smallmatrix}\right)$$

$$\left(\begin{smallmatrix} \text{---} & \circ\Box\circ \\ \text{---} & \end{smallmatrix}\right) \sim \left(\begin{smallmatrix} \perp & \perp \\ \perp & \end{smallmatrix}\right)$$

$$\left(\begin{smallmatrix} \text{---} & \circ\Box\circ \\ \text{---} & \end{smallmatrix}\right) \sim \left(\begin{smallmatrix} \perp & \perp \\ \perp & \end{smallmatrix}\right)$$

$$\left(\begin{smallmatrix} \text{---} & \circ\Box\circ \\ \text{---} & \end{smallmatrix}\right) \sim \left(\begin{smallmatrix} \perp & \perp \\ \perp & \end{smallmatrix}\right)$$

Q. Why does Firefox do this?

A. bfcache.

Q. bfcache?

A. 'back/forward cache'.

Q. Since when?

A. Firefox 1.5



Classic problem #3

```
someDomElement.append(htmlString);
```

What happens when the user goes 'back'?

- Chrome and IE will not preserve the DOM, all non-serverside DOM markup/manipulations will be lost

What happens when the user goes 'back'?

- Chrome and IE will not preserve the DOM, all non-serverside DOM markup/manipulations will be lost
- Firefox's bfcache will preserve the DOM, including all non-server side DOM manipulations

What happens when the user goes 'back'?

- Chrome and IE will not preserve the DOM, all non-serverside DOM markup/manipulations will be lost
- Firefox's bfcache will preserve the DOM, including all non-server side DOM manipulations
- Safari - it's complicated.

$$\left(\begin{smallmatrix} \text{---} & \circ\Box\circ \\ \text{---} & \end{smallmatrix}\right) \sim \left(\begin{smallmatrix} \perp & \perp \\ \perp & \end{smallmatrix}\right)$$

$$\left(\begin{smallmatrix} \text{---} & \circ\Box\circ \\ \text{---} & \end{smallmatrix}\right) \sim \left(\begin{smallmatrix} \perp & \perp \\ \perp & \end{smallmatrix}\right)$$

$$\left(\begin{smallmatrix} \text{---} & \circ\Box\circ \\ \text{---} & \end{smallmatrix}\right) \sim \left(\begin{smallmatrix} \perp & \perp \\ \perp & \end{smallmatrix}\right)$$

$$\left(\begin{smallmatrix} \text{---} & \circ\Box\circ \\ \text{---} & \end{smallmatrix}\right) \sim \left(\begin{smallmatrix} \perp & \perp \\ \perp & \end{smallmatrix}\right)$$

What are some solutions?

HTTP headers - a not so subtle tactic for the redirect issue

Cache-Control: no-cache, max-age=0, must-revalidate,
no-store

setTimeout - yeah I know.

```
<script>
  window.onload = function() {
    setTimeout(document.location = 'another_page.html', 10)
  }
</script>
```

bfcache solution - with caveats

```
<body onload="myWholePagesFunctionsWrapper()">
```

another bfcache solution - still not fabulous

```
window.onunload = function(){};
```

`onload()` and `onunload()`
circumvent the point of
`bfcache`

just write your code to be mindful of the differences in history caching

- watch any flags/counters you set in JavaScript
- make use of url hashes if you need more control
- only bust the cache if you absolutely have to

... what about React?

you might wanna look at

github.com/leifdenby/jshistoryreactmixin

nice.

TT)(° - °))