

Offline is the new Black

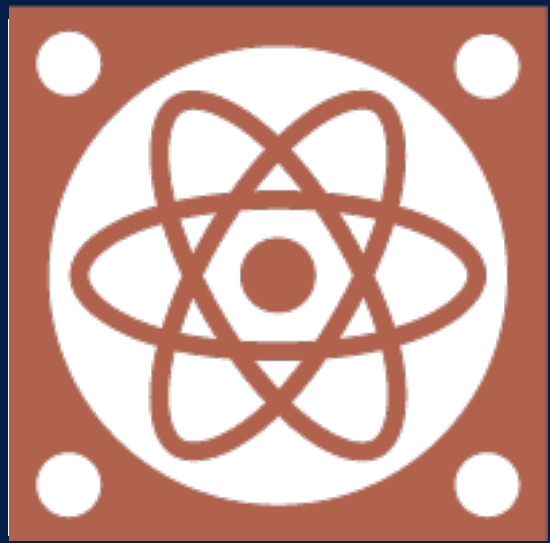
Max Stoiber

@mxstbr

Open Source Developer, Thinkmill



Thinkmill

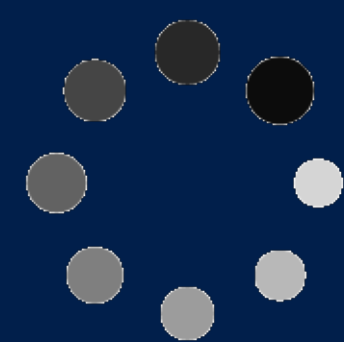






2 hours of electricity per day

2 hours of network per day



Network coverage isn't there

The web is information

Be there when your users need
you

AppCache

“AppCache is a douchebag”

Jake Archibald, mxs.is/googne



[Home](#) › [Documents](#) › [Firefox 44](#)

Application Cache API has been deprecated

Published: September 25, 2015 | Categories: [HTML](#)

Description

The HTML5 [Application Cache API](#) (AppCache), that enables [offline support](#) in Web applications, is now considered deprecated and will be [removed from a future Firefox release](#). Web developers are encouraged to learn [Service Workers](#) as an alternative solution, so that you can offer a richer, more seamless offline user experience sometime soon. This new API is already available on [Firefox Developer Edition](#) for testing, though not all the features are implemented yet. Note that AppCache

[blink-dev](#) ›

Intent to deprecate (and then remove) AppCache on insecure origins

63 Einträge von 19 Autoren

**Joel Weinberger**

8. Jan.

[Nachricht auf Deutsch übersetzen](#)

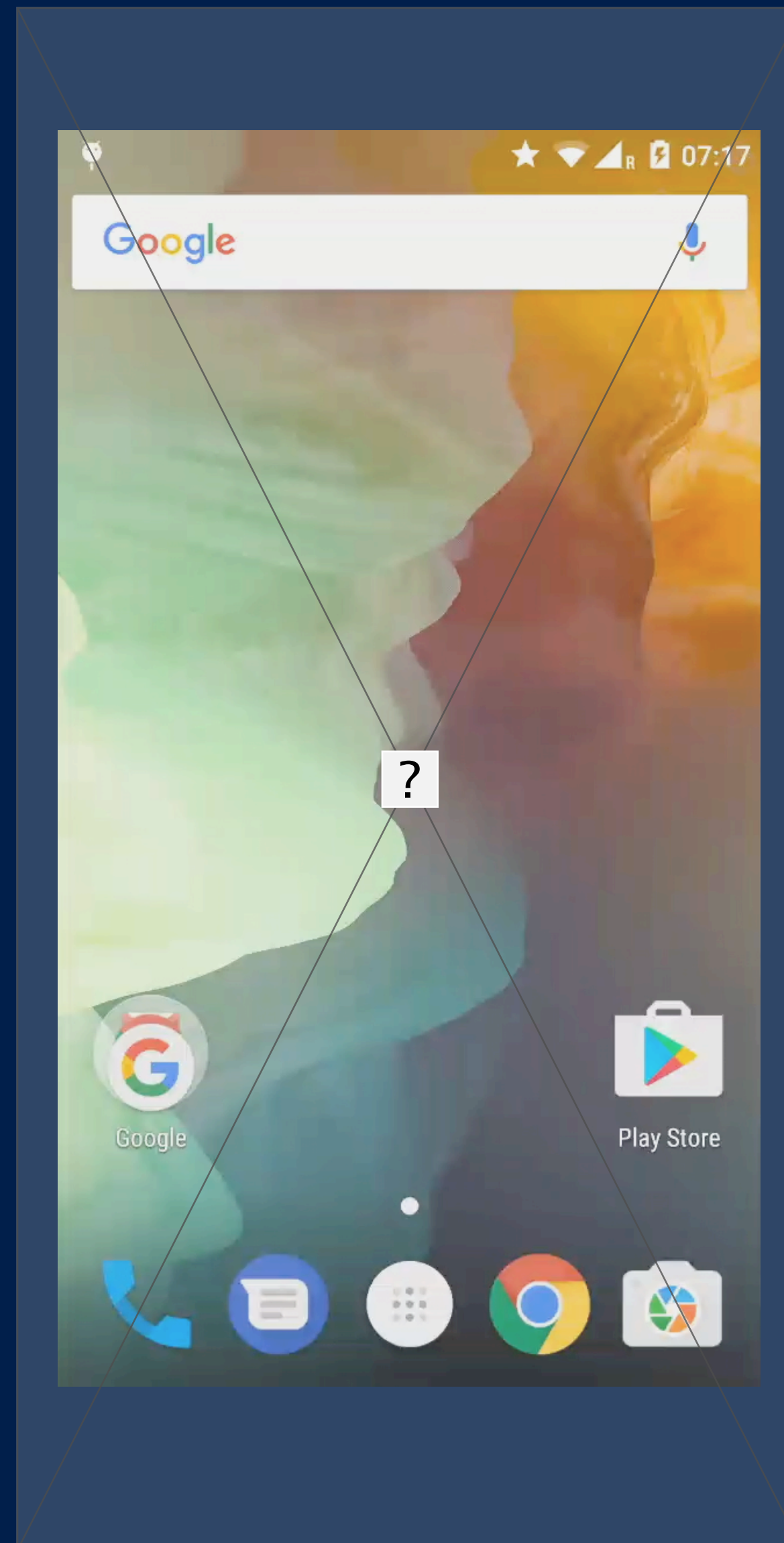
Firefox recently announced an intent to remove AppCache entirely: <https://www.fxsitecompat.com/en-US/docs/2016/application-cache-support-will-be-removed/>

While it seems awfully aggressive to straight up remove AppCache immediately, it has been a long source of consternation to the security team that AppCache is allowed over insecure origins. This allows for potentially persistent, online and offline, XSS of sites, which is a pretty serious privilege escalation from regular XSS.

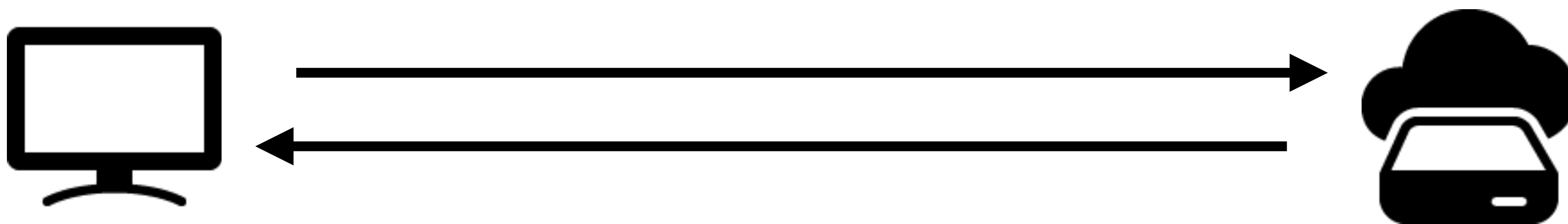
In the spirit of [Deprecating Powerful Features on Insecure Origins](#), we are proposing to deprecate and then remove AppCache on insecure origins. I'd like to add a deprecation message ASAP, and then in a release or two, do the actual disabling.

Unfortunately, we don't have perfect numbers for usage at the moment, since there isn't a proper WebCore.FeatureObserver entry for AppCache. However, doing some rough math using WebCore.FeatureObserver.PageVisits and AppCache.MainPageLoad entries, it appears that around 1.9% of all page loads use include an AppCache main page load event, but only 0.05% do that over an insecure origin. While not quite at the 0.03% level we'd like it to be, I'd also point out

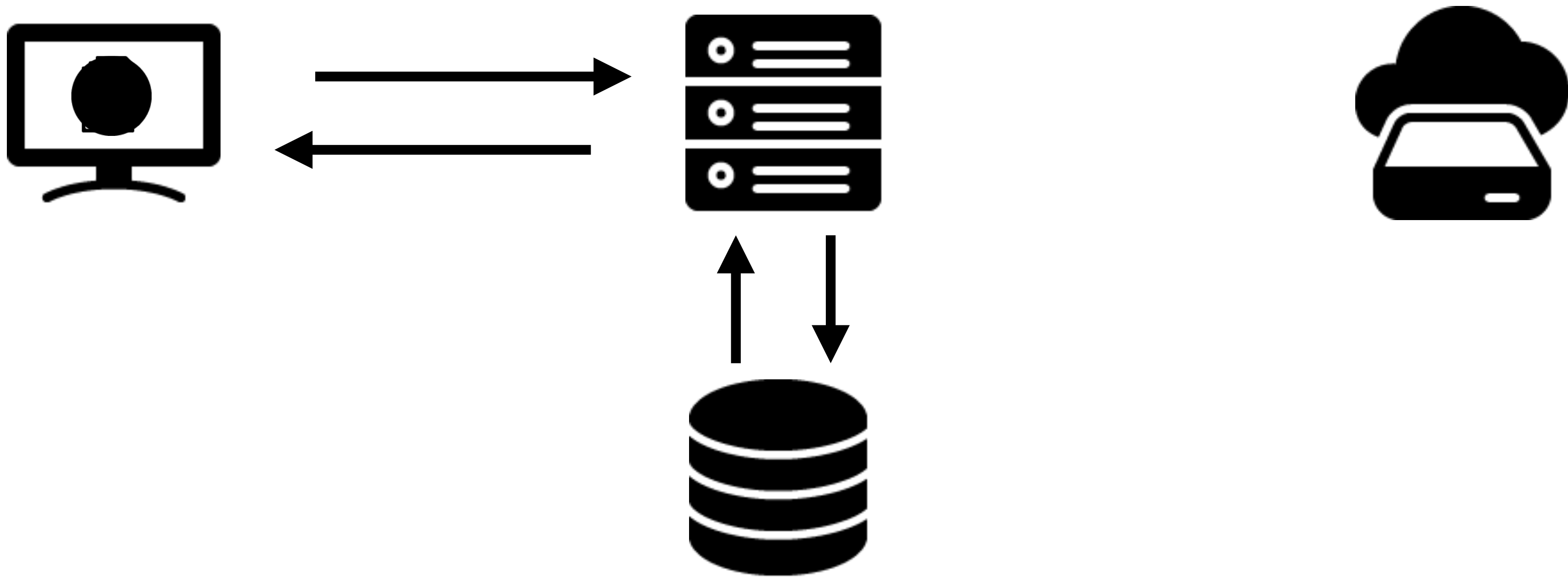
 ServiceWorker 

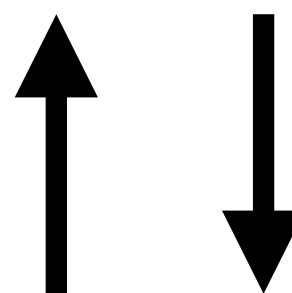
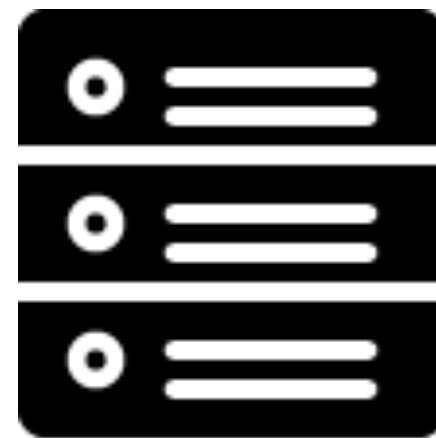


mxs.is/asdk










```
1  if ('serviceWorker' in navigator) {  
2      navigator.serviceWorker.register('serviceworker.js')  
3          .then(function() {  
4              console.log('ServiceWorker registration successful!');  
5          }).catch(function(err) {  
6              console.log('ServiceWorker registration failed:', err);  
7          });  
8  }
```



```
1  var CACHE_NAME = 'demo-cache-v2';
2  var urlsToCache = [
3      'index.html',
4      'bundle.js',
5      'style.css',
6      'debugging.gif'
7  ];
8
9  self.addEventListener('install', function(event) {
10      caches.open(CACHE_NAME)
11          .then(function(cache) {
12              console.log('installed, add all files to cache');
13              return cache.addAll(urlsToCache);
14          });
15  });
16
```

```
1  var CACHE_NAME = 'demo-cache-v2';
2  var urlsToCache = [
3    'index.html',
4    'bundle.js',
5    'style.css',
6    'debugging.gif'
7  ];
8
9  self.addEventListener('install', function(event) {
10    caches.open(CACHE_NAME)
11      .then(function(cache) {
12        console.log('installed, add all files to cache');
13        return cache.addAll(urlsToCache);
14      });
15  });
16
```

```
1  var CACHE_NAME = 'demo-cache-v2';
2  var urlsToCache = [
3    'index.html',
4    'bundle.js',
5    'style.css',
6    'debugging.gif'
7  ];
8
9  self.addEventListener('install', function(event) {
10     caches.open(CACHE_NAME)
11       .then(function(cache) {
12         console.log('installed, add all files to cache');
13         return cache.addAll(urlsToCache);
14       });
15   });
16
```


29

```
return fetch(fetchRequest).then(
```

30

```
function(response) {
```

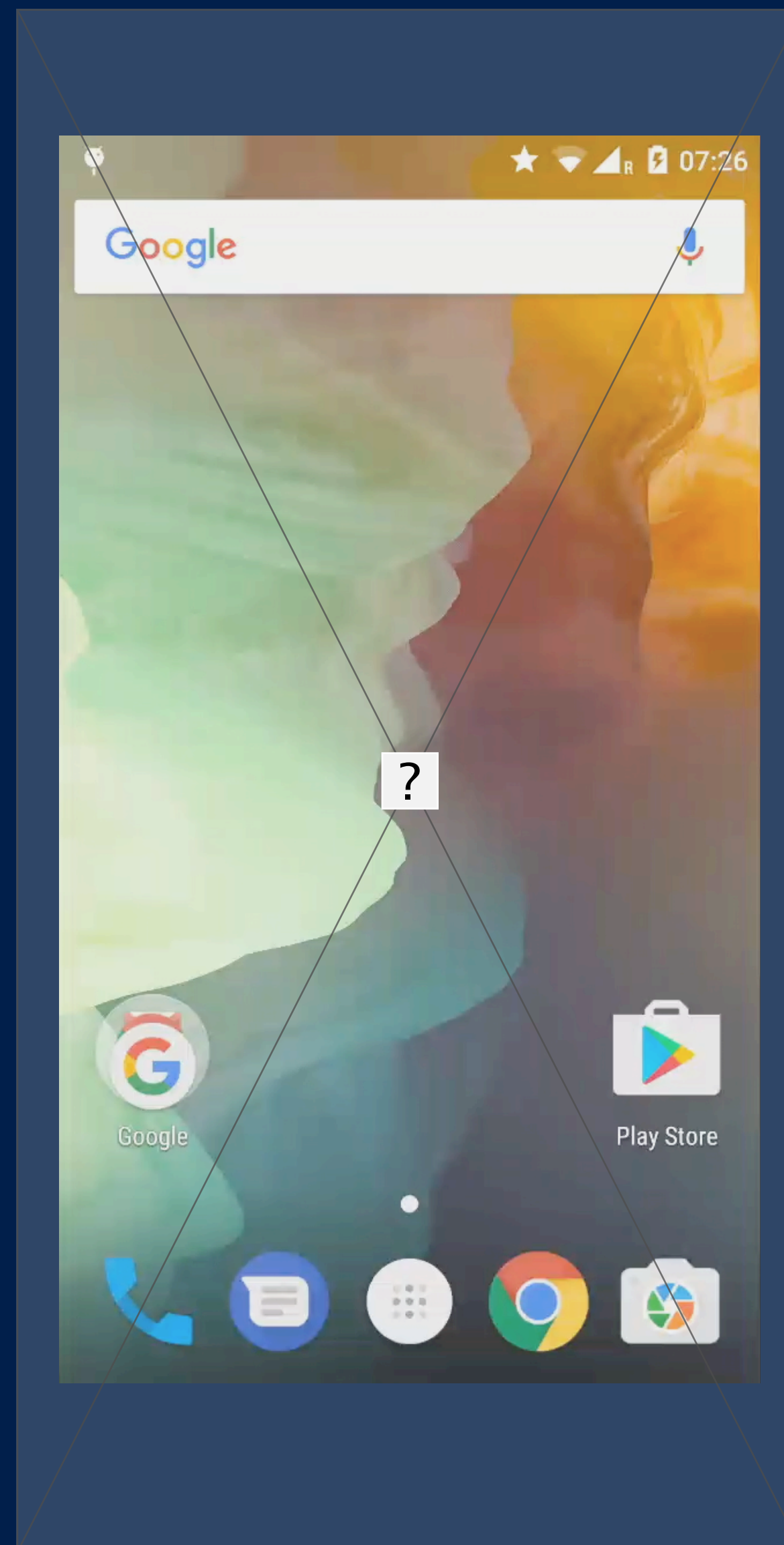
```
26  
27 var fetchRequest = event.request.clone();  
28  
29 return fetch(fetchRequest).then(  
30   function(response) {  
31     console.log('get files from server');  
32     if(!response || response.status !== 200) {  
33       console.log('server error');  
34       return response;  
35     }  
36  
37     var responseToCache = response.clone();  
38  
39     caches.open(CACHE_NAME)  
40       .then(function(cache) {  
41         console.log('put new files in cache').
```

```
36
37     var responseToCache = response.clone();
38
39     caches.open(CACHE_NAME)
40       .then(function(cache) {
41         console.log('put new files in cache');
42         cache.put(event.request, responseToCache);
43       });
44     console.log('return server files');
45     return response;
46   }
47 );
48 })
49 );
50 });
51
```

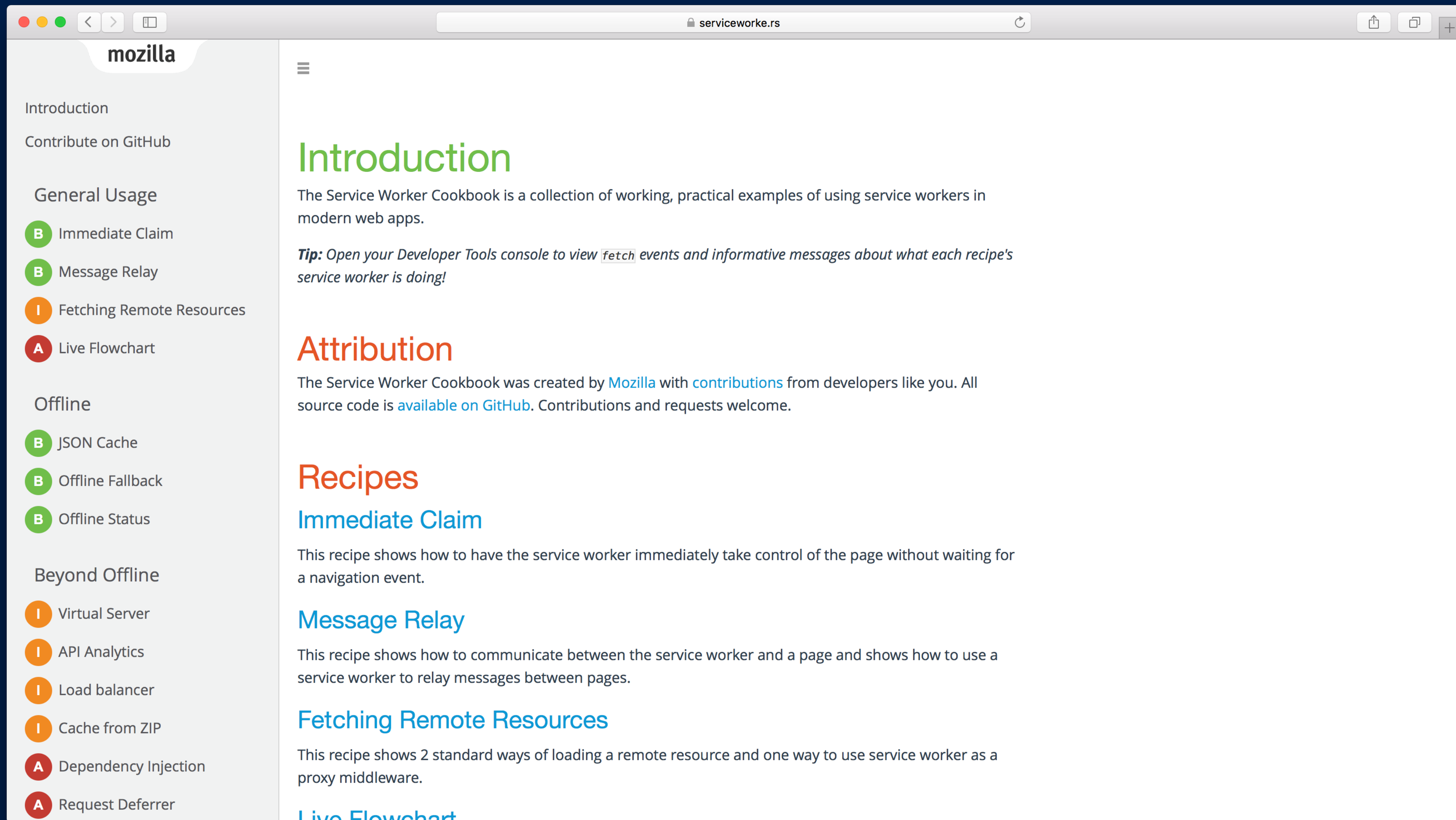


```
36
37     var responseToCache = response.clone();
38
39     caches.open(CACHE_NAME)
40     .then(function(cache) {
41         console.log('put new files in cache');
42         cache.put(event.request, responseToCache);
43     });
44     console.log('return server files');
45     return response;
46 }
47 );
48 })
49 );
50 });
51
```

```
36
37     var responseToCache = response.clone();
38
39     caches.open(CACHE_NAME)
40     .then(function(cache) {
41         console.log('put new files in cache');
42         cache.put(event.request, responseToCache);
43     });
44     console.log('return server files');
45     return response;
46 }
47 );
48 })
49 );
50 });
51
```

mxs.is/asdk



serviceworker.rs

caniuse.com

About

News

May 12, 2015 - Feature suggestions now handled by GitHub issues

Compare browsers

Index

AdChoices

HTML5 CSS3

HTML5 Browser

Web Notifications

CSS3 Class

Can I use

Service Workers

? Settings

1 result found

#

Service Workers

- WD

Global

0% + 53.31% = 53.31%

Method that enables applications to take advantage of persistent background processing, including hooks to enable bootstrapping of web applications while offline.

Current aligned

Usage relative

Show all

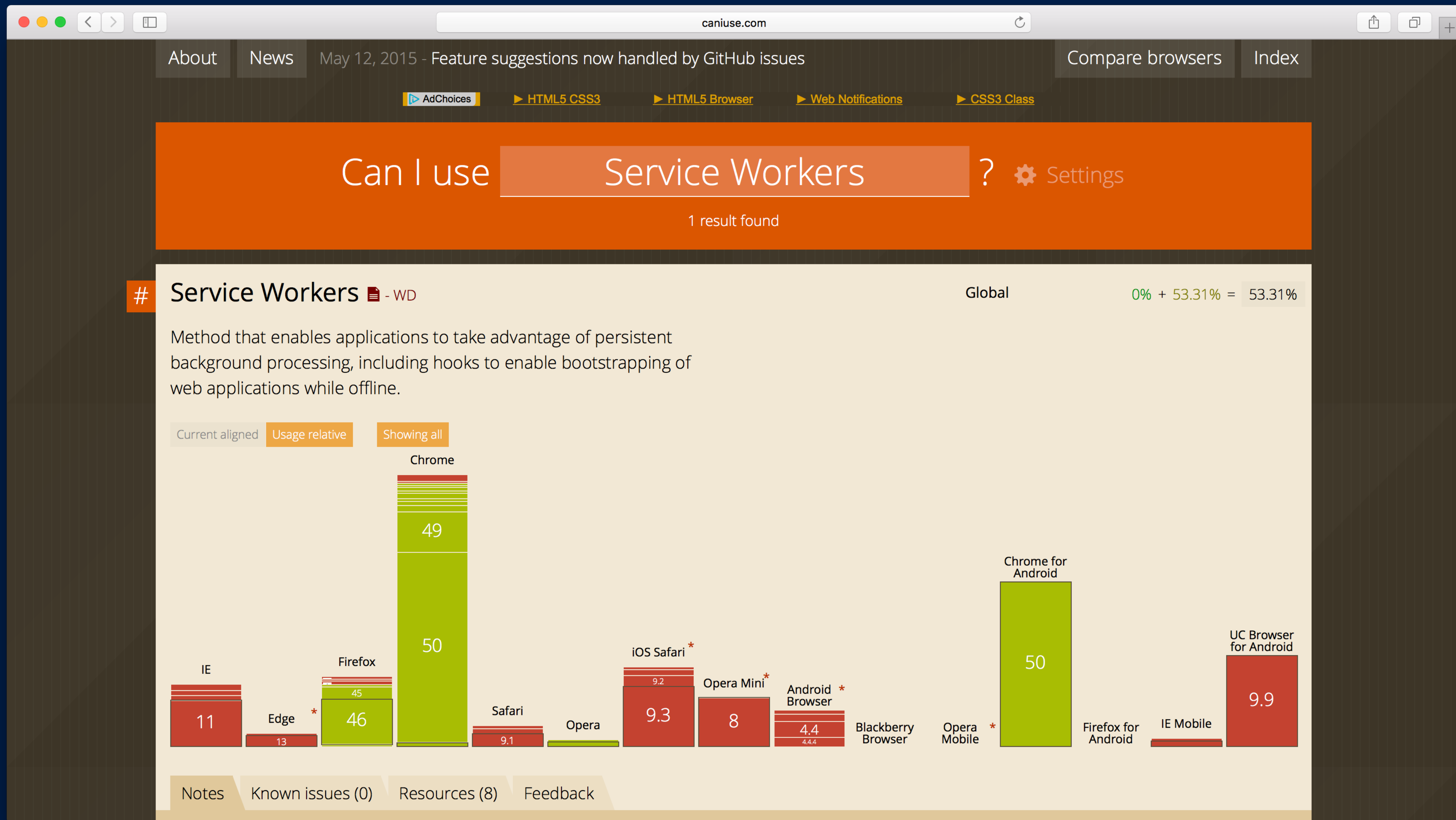
IE	Edge	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Chrome for Android
			29						
			45						
			48					4.3	
			49			8.4		4.4	
8		45	50			9.2		4.4.4	
11	13	46	51	9.1	37	9.3	8	50	50
	14	47	52	TP	38				
		48	53		39				
		49	54						

Notes

Known issues (0)

Resources (8)

Feedback



Manifest


```
ServiceWorker Demo

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1">
6    <link rel="manifest" href="manifest.json">
7    <title>ServiceWorker Demo</title>
8  </head>
9  <body>
10   <div>
11     <h1>Hello World! 🙌</h1>
12     <h2>What debugging feels like:</h2>
13     
14   </div>
15   <link rel="stylesheet" href="style.css">
16   <script src="bundle.js"></script>
17 </body>
18 </html>
19
```

ServiceWorker Demo

```
1  {
2    "name": "Demo App",
3    "icons": [{
4      "src": "wave.png",
5      "sizes": "256x256",
6      "type": "image/png"
7    }],
8    "start_url": "index.html",
9    "display": "standalone",
10   "orientation": "portrait"
11 }
12
```

ServiceWorker Demo

```
1  {
2    "name": "Demo App",
3    "icons": [{
4      "src": "wave.png",
5      "sizes": "256x256",
6      "type": "image/png"
7    }],
8    "start_url": "index.html",
9    "display": "standalone",
10   "orientation": "portrait"
11 }
12
```


ServiceWorker Demo

```
1  {
2    "name": "Demo App",
3    "icons": [{
4      "src": "wave.png",
5      "sizes": "256x256",
6      "type": "image/png"
7    }],
8    "start_url": "index.html",
9    "display": "standalone",
10   "orientation": "portrait"
11 }
12
```

ServiceWorker Demo

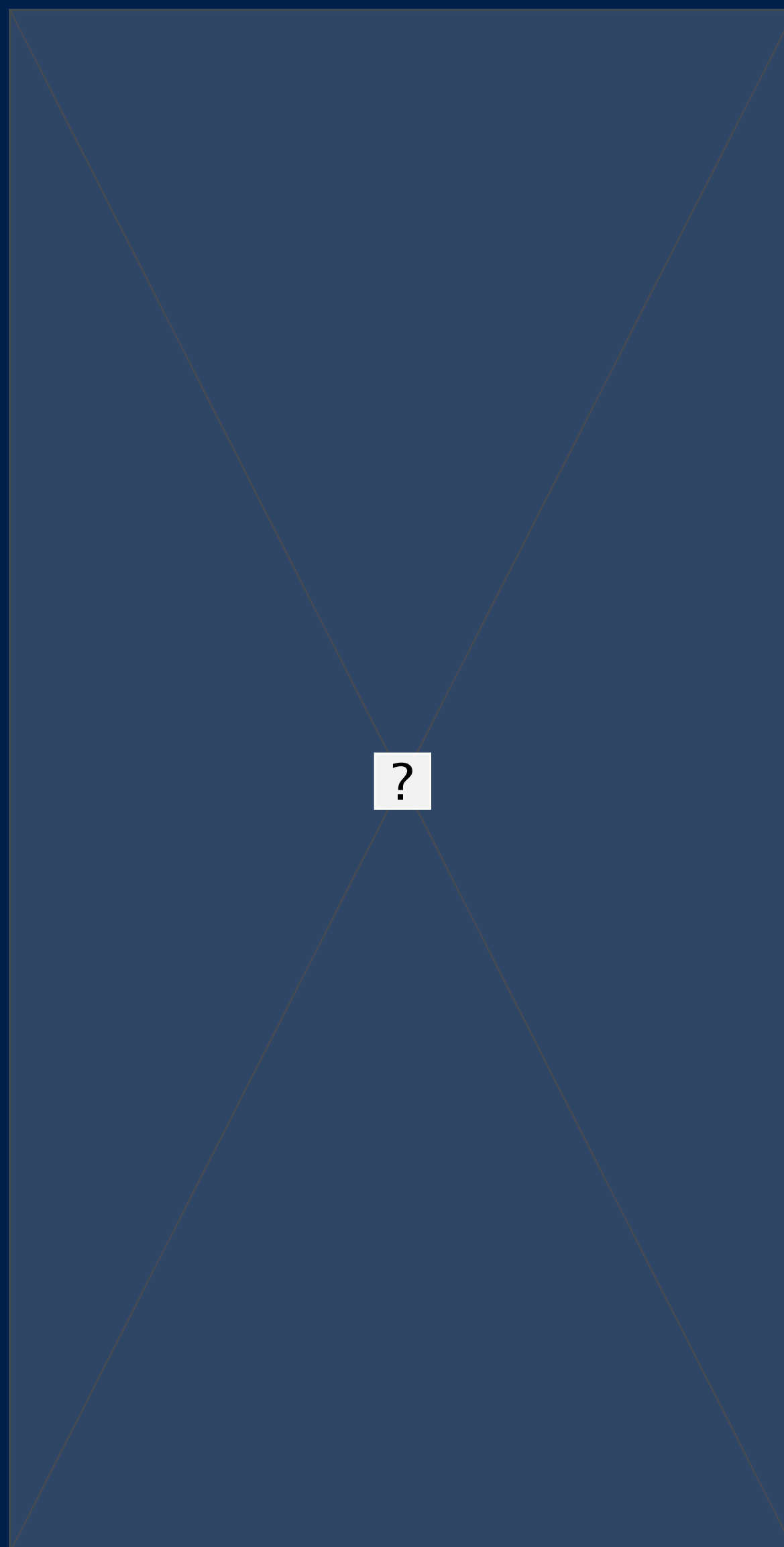
```
1  {
2    "name": "Demo App",
3    "icons": [{
4      "src": "wave.png",
5      "sizes": "256x256",
6      "type": "image/png"
7    }],
8    "start_url": "index.html",
9    "display": "standalone",
10   "orientation": "portrait"
11  }
12
```


ServiceWorker Demo

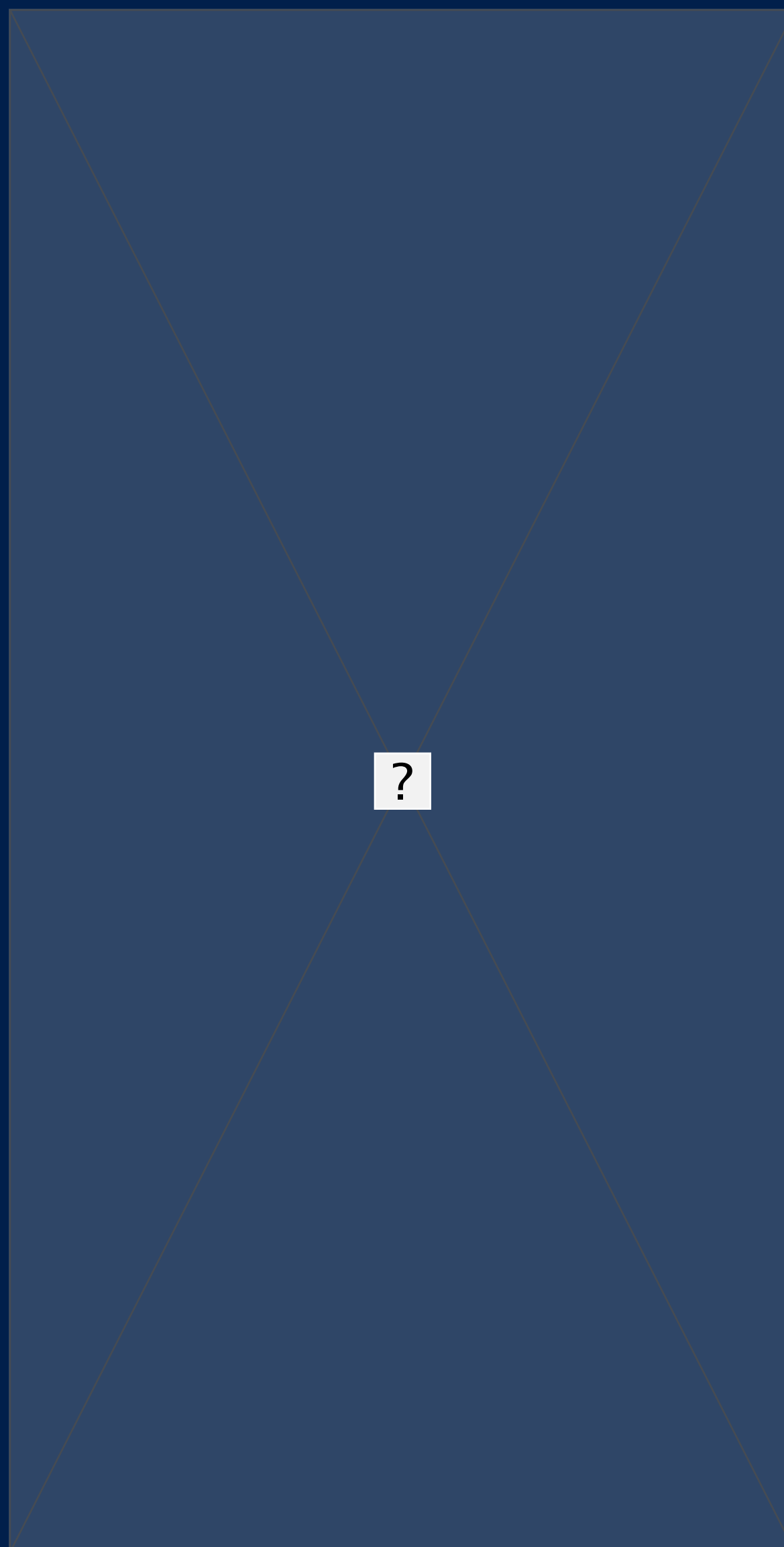
```
1  {
2    "name": "Demo App",
3    "icons": [{
4      "src": "wave.png",
5      "sizes": "256x256",
6      "type": "image/png"
7    }],
8    "start_url": "index.html",
9    "display": "standalone",
10   "orientation": "portrait"
11  }
12
```

ServiceWorker Demo

```
1  {
2    "name": "Demo App",
3    "icons": [{
4      "src": "wave.png",
5      "sizes": "256x256",
6      "type": "image/png"
7    }],
8    "start_url": "index.html",
9    "display": "standalone",
10   "orientation": "portrait"
11 }
12
```



mxs.is/asdk



mxs.is/asdk

```
1  // \ serviceworker in navigator / {
2  navigator.serviceWorker.register('serviceworker.js')
3    .then(function() {
4      console.log('ServiceWorker registration successful!');
5    }).catch(function(err) {
6      console.log('ServiceWorker registration failed:', err);
7    });
8
9  navigator.serviceWorker.addEventListener('controllerchange', function(event) {
10    navigator.serviceWorker.controller.addEventListener('statechange', function() {
11      if (this.state === 'activated') {
12        var message = document.querySelector('.offline-message');
13        message.classList.remove('hidden');
14        setTimeout(function () {
15          message.classList.add('hidden');
16        }, 5000);
17      }
18    });
19  });
```



```
1  // \ serviceworker in navigator / {
2  navigator.serviceWorker.register('serviceworker.js')
3    .then(function() {
4      console.log('ServiceWorker registration successful!');
5    }).catch(function(err) {
6      console.log('ServiceWorker registration failed:', err);
7    });
8
9  navigator.serviceWorker.addEventListener('controllerchange', function(event) {
10    navigator.serviceWorker.controller.addEventListener('statechange', function() {
11      if (this.state === 'activated') {
12        var message = document.querySelector('.offline-message');
13        message.classList.remove('hidden');
14        setTimeout(function () {
15          message.classList.add('hidden');
16        }, 5000);
17      }
18    });
19  });
```

```
1  // \ serviceworker in navigator / {
2  navigator.serviceWorker.register('serviceworker.js')
3    .then(function() {
4      console.log('ServiceWorker registration successful!');
5    }).catch(function(err) {
6      console.log('ServiceWorker registration failed:', err);
7    });
8
9  navigator.serviceWorker.addEventListener('controllerchange', function(event) {
10    navigator.serviceWorker.controller.addEventListener('statechange', function() {
11      if (this.state === 'activated') {
12        var message = document.querySelector('.offline-message');
13        message.classList.remove('hidden');
14        setTimeout(function () {
15          message.classList.add('hidden');
16        }, 5000);
17      }
18    });
19  });
```

```
1  // \ serviceworker in navigator / {
2  navigator.serviceWorker.register('serviceworker.js')
3    .then(function() {
4      console.log('ServiceWorker registration successful!');
5    }).catch(function(err) {
6      console.log('ServiceWorker registration failed:', err);
7    });
8
9  navigator.serviceWorker.addEventListener('controllerchange', function(event) {
10    navigator.serviceWorker.controller.addEventListener('statechange', function() {
11      if (this.state === 'activated') {
12        var message = document.querySelector('.offline-message');
13        message.classList.remove('hidden');
14        setTimeout(function () {
15          message.classList.add('hidden');
16        }, 5000);
17      }
18    });
19  });
```



```
1  // \ serviceworker in navigator / {
2  navigator.serviceWorker.register('serviceworker.js')
3    .then(function() {
4      console.log('ServiceWorker registration successful!');
5    }).catch(function(err) {
6      console.log('ServiceWorker registration failed:', err);
7    });
8
9  navigator.serviceWorker.addEventListener('controllerchange', function(event) {
10    navigator.serviceWorker.controller.addEventListener('statechange', function() {
11      if (this.state === 'activated') {
12        var message = document.querySelector('.offline-message');
13        message.classList.remove('hidden');
14        setTimeout(function () {
15          message.classList.add('hidden');
16        }, 5000);
17      }
18    });
19  });
```



```
1  // \ serviceworker in navigator / {
2  navigator.serviceWorker.register('serviceworker.js')
3    .then(function() {
4      console.log('ServiceWorker registration successful!');
5    }).catch(function(err) {
6      console.log('ServiceWorker registration failed:', err);
7    });
8
9  navigator.serviceWorker.addEventListener('controllerchange', function(event) {
10    navigator.serviceWorker.controller.addEventListener('statechange', function() {
11      if (this.state === 'activated') {
12        var message = document.querySelector('.offline-message');
13        message.classList.remove('hidden');
14        setTimeout(function () {
15          message.classList.add('hidden');
16        }, 5000);
17      }
18    });
19  });
```



```
39     .then(function(cache) {
40         cache.put(event.request, responseToCache);
41     });
42     console.log('[SW] Return new files to client');
43     return response;
44 });
45 })
46 );
47 });
```

```
48
49 self.addEventListener('activate', function(event) {
50     console.log('[SW] Claiming this ServiceWorker!');
51     event.waitUntil(self.clients.claim());
52 });
```

```
53
```

```
54
```

```
55
```

```
56
```

```
57
```

```
39     .then(function(cache) {
40         cache.put(event.request, responseToCache);
41     });
42     console.log('[SW] Return new files to client');
43     return response;
44 });
45 })
46 );
47 });
```

```
48
49 self.addEventListener('activate', function(event) {
50     console.log('[SW] Claiming this ServiceWorker!');
51     event.waitUntil(self.clients.claim());
52 });
```

```
53
```

```
54
```

```
55
```

```
56
```

```
57
```



mxs.is/asdk

Thank you!

Max Stoiber

@mxstbr

Come talk to me!