Offline First

Martin Hilscher
JUNGE HAIE

Offline First – say what?

- UX-Design Strategie/Philosophie
- (Web-)Anwendungen voll funktionsfähig …
- … auch ohne Netz


wat

Fallacies of distributed computing

1. The network is reliable
2. Latency is zero
3. Bandwidth is infinite
4. The network is secure
5. Topology doesn't change
6. There is one administrator
7. Transport cost is zero
8. The network is homogeneous

The network is reliable

Latency is zero

L1 cache	0.5 ns	
L2 cache	7 ns	14x
Main memory	100 ns	20x
1 Block von SSD lesen	150,000 ns	1500x
Ping CA → NL	150,000,000 ns	1000x

SSD-Zugriff fällt oft schon negativ auf

Bandwidth is infinite

Wie krieg ich am schnellsten 2.5TB Daten von Panama nach München?

Technologien

Was brauchen wir

- Kontrolle über das Caches
- Kontrolle über HTTP(S)-Anfragen

Clientside Storage

Cookies

Pro

Funtkioniern überall

Con

... es sei denn, der Benutzer blocked Cookies

Interface etwas hakelig

Werden vollständig in den Arbeitsspeicher geladen

Wieviel Speicher darf ich eignetlich benutzen?

...

⇒ Werde ich nicht weiter betrachten

Localstorage/Sessionstorage

- Auch DOM-Storage oder Supercookies genannt
- Key-Value Store
- localStorage persistent
- sessionStorage nur pro Session

Pros/Cons

Pros	Cons
Key-Value Store	Kein natives Indexing
Session Konzept eingebaut	Nur 5MB Speicher
Breite Unterstüzung	

Gut zu benutzen, wenn euer Szenario mit den Limitierungen klar kommt.

Can I use?


Web Storage - name/value pairs - LS
Global
90.42% + 0.05% = 90.47%
Method of storing data locally like cookies, but for larger amounts of data (sessionStorage and localStorage, used to fall under HTML5).
IE
Edge
Firefox
Chrome
Safari
Opera
iOS Safari
Opera Mini
Android Browser
Chrome for Android
8
9
11
13
14
44
45
46
47
48
47
48
49
50
51
52
53
9
9.1
TP
36
37
38
8.4
9.2
9.3
8
4.3
4.4
4.4.4
47
49
Notes
Known issues (8)
Resources (8)
Feedback
No notes

WebSQL

- Zugriff auf SQLite in Browser
- "Most probably the worst implemented piece of SHIT …"
- Wird nicht mehr weiterentwickelt
- Wird in Zukunft gedropped

Can I use?


Web SQL Database - UNOFF
Global
72.56%
Method of storing data client-side, allows Sqlite database queries for access and manipulation
IE
Edge
Firefox
Chrome
Safari
Opera
iOS Safari
Opera Mini
Android Browser
Chrome for Android
8
9
11
13
14
44
45
46
47
48
47
48
49
50
51
52
53
9
9.1
TP
36
37
38
8.4
9.2
9.3
8
4.3
4.4
4.4.4
47
49
Notes
Known issues (2)
Resources (5)
Feedback
The Web SQL Database specification is no longer being maintained and support may be dropped in future versions.
MS Edge status: Not currently planned

Was würden wir uns wünschen?

- Key-Value Store
- Mit Index
- Möglichst groß

Say hello to IndexDB

Pros / Cons

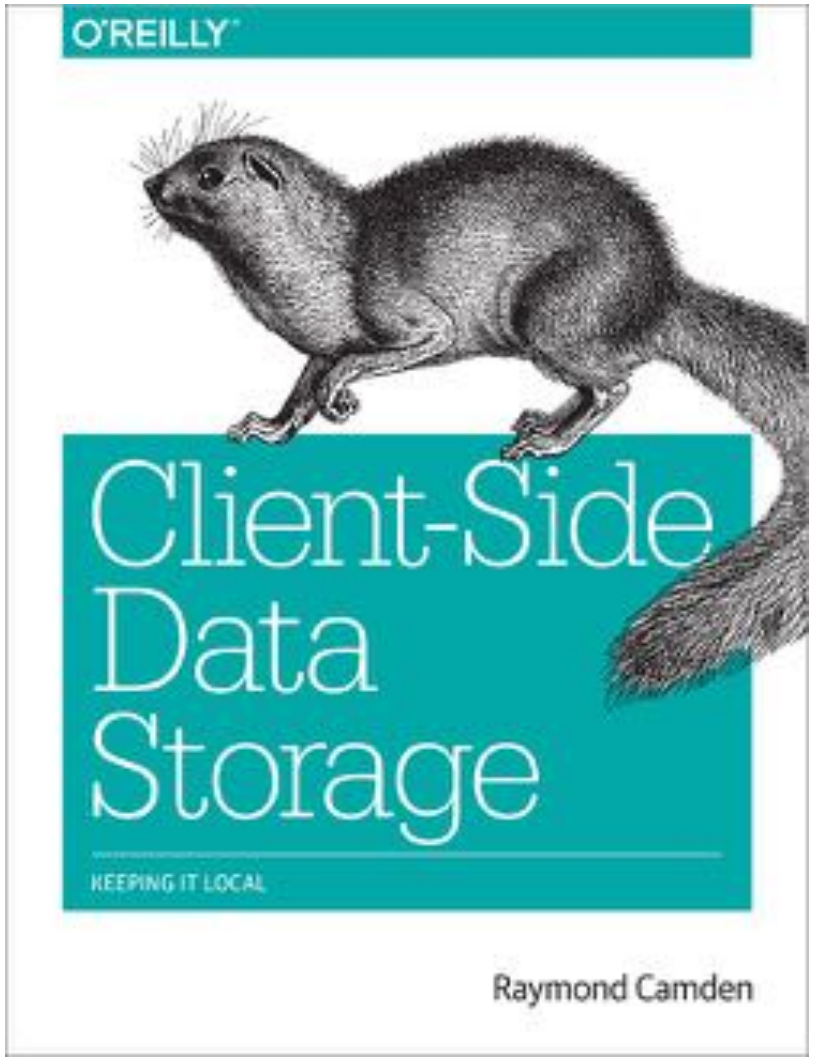
Pros	Cons
Key-Value store	API so meh :-(
Persistent	
Mit Indexen	

Bis auf die API genau was wir wollen

Can I use?


IndexedDB - REC
Method of storing data client-side, allows indexed database queries.
Global 59.7% + 18.12% = 77.83%
unprefixed 59.59% + 17.88% = 77.47%
IE
Edge
Firefox
Chrome
Safari
Opera
iOS Safari
Opera Mini
Android Browser
Chrome for Android
8
9
11
13
14
44
45
46
47
48
47
48
49
50
51
52
53
9
9.1
TP
36
37
38
8.4
9.2
9.3
8
4.3
4.4
4.4.4
47
49
Notes
Known issues (3)
Resources (9)
Feedback
1 Partial support in IE 10 & 11 refers to a number of subfeatures not being supported.
2 Partial support in iOS 8 & 9 refers to seriously buggy behavior as well as complete lack of support in WebViews.

Further read



http://shop.oreilly.com/product/0636920043676.do

PouchDB

- Key-Value Store
- Abstrahiert WebSQL bzw. IndexDB
- Implementiert die API von CouchDB
- … einschließlich des Replikations-Algorithmus
- … und changes feed

Demo

Fine Grained HTTP Control

Appcache

Wir reden nicht über Appcache

"AppCache broke a lot of eggs, and failed to make an Omlet"
Jake Archibald (Coauthor der AppCache Spec)

Hear for yourselves

Servicewoker

Serviceworker

- Entwickelt von Google, Firefox & Samsung
- Fängt alle HTTP calls App
- ... und liefert API sie zu verändern

Register

```
navigator.serviceWorker.register('/worker.js').then(function(reg) {
  console.log(":-)", reg);
}, function (err) {
  console.log(":-(", err)
});
```

Routing

```
self.addEventListener('fetch'), function(event) {
  event.respondWith(
    new Response('<h1>Sorry</h1>Nothing to see here, {
                headers: { 'Content-Type': 'text/html'}
  });
});
```

Caching

```
self.addEventListener('install', function (event) {
        event.waitUntil(
                caches.create('cache').then(function(cache){
                        return ache.addAll([
                                '/staticAssets/logo.svg',
                                '/staticAssets/icons.svg',
                                '/js/vendor.js',
                                '/js/all.js',
                                '/styles/style.css'
                        ]);
                })
        );
});
```

Caching

```
self.addEventListener('fetch', function (event) {
        event.respondWith(caches.match(event.request));
});
```

Caching

```
self.addEventListener('fetch', function (event) {
        event.respondWith(caches.match(event.request))
        .then (function(response) {
                return response || caches.match('fallback.html');
        });
});
```

Caching

```
self.addEventListener('fetch', function (event) {
        event.respondWith(caches.match(event.request))
        .then (function(response) {
                return response || fetch(event.request);
        });
});
```

Caching

```
self.addEventListener('fetch', function(event) {
  event.respondWith(
    caches.open('cache').then(function(cache) {
      return cache.match(event.request).then(function (response) {
        return response || fetch(event.request).then(function(response) {
          cache.put(event.request, response.clone());
          return response;
        });
      });
    })
  );
});
```

Caching

```
self.addEventListener('fetch', function(event) {
  event.respondWith(
    caches.open('cache').then(function(cache) {
      return cache.match(event.request).then(function(response) {
        var fetchPromise = fetch(event.request).then(function(networkResponse) {
          cache.put(event.request, networkResponse.clone());
          return networkResponse;
        })
        return response || fetchPromise;
      })
    })
  );
});
```

… and more

ServiceWorker können noch mehr …

- … auf PushNotifications regieren
- … periodisch im Hintergrund updaten

Can I use?


Service Workers - WD
Global
0% + 53.5% = 53.5%
Method that enables applications to take advantage of persistent background processing, including hooks to enable bootstrapping of web applications while offline.
Show all
IE
Edge
Firefox
Chrome
Safari
Opera
iOS Safari
Opera Mini
Android Browser
Chrome for Android
8
9
11
13
14
44
45
46
47
48
47
48
49
50
51
52
53
9
9.1
TP
36
37
38
8.4
9.2
9.3
8
4.3
4.4
4.4.4
47
49
Notes
Known issues (0)
Resources (8)
Feedback
Details on partial support can be found on Is ServiceWorker Ready?
MS Edge status: Under Consideration
WebKit status: Under Consideration
1 Partial support can be enabled in Firefox with the dom.serviceWorkers.enabled flag.

UI/UX

UI/UX

- Sync / Save / ...
- Icons (Diskette für Save)
- Wie mache ich dem_der Nutzer*in klar, das alles gesichert ist
- Wie mach ich dem_der Nutzer*in klar das alles gesynct ist
- Wie füge ich bei zeitlich sortierten Dingen neue Inhalte ein

Discuss