Offline strategies for HTML5 web applications

Stephan Hochdörfer, bitExpert AG

About me

- Stephan Hochdörfer
- Head of IT at bitExpert AG, Germany
- enjoying PHP since 1999
- S.Hochdoerfer@bitExpert.de
- @shochdoerfer

Storing data on the client? Seriously?

Offline strategies for HTML5 web applications

How did we solve these issues in the past?

Cookies are tasty, but not awesome!

IE DHTML behaviours, not sweet!

Offline strategies for HTML5 web applications

Flash Cookies are yummie!

Google Gears made it right. Sort of.

Offline strategies for HTML5 web applications

Can I haz alternative?

HTML

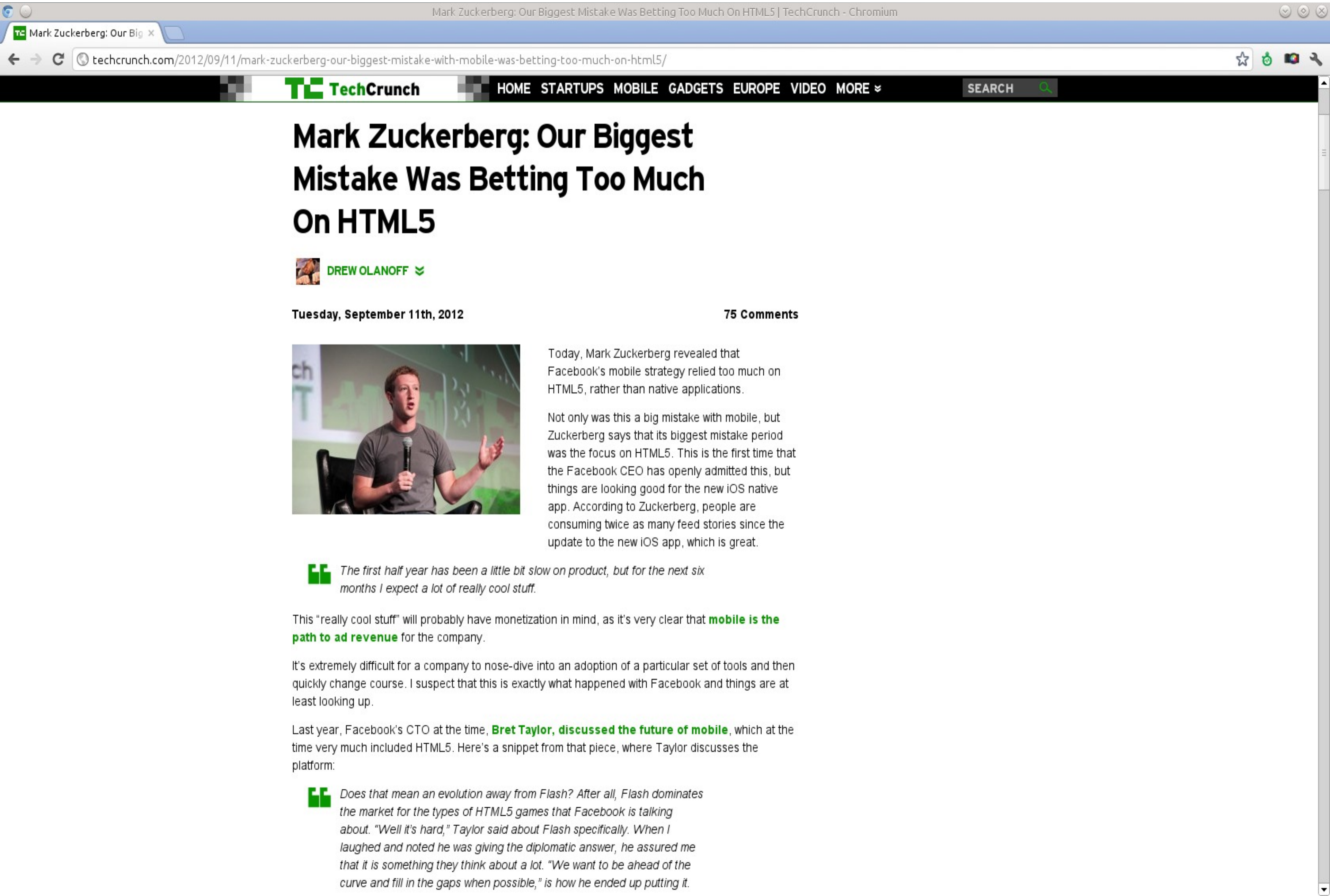
5

to the rescue!

[...] we take the next step, announcing 2014 as the target for Recommendation.

Jeff Jaffe, Chief Executive Officer, World Wide Web Consortium

Offline strategies for HTML5 web applications



Mark Zuckerberg: Our Biggest Mistake Was Betting Too Much On HTML5 | TechCrunch - Chromium

techcrunch.com/2012/09/11/mark-zuckerberg-our-biggest-mistake-with-mobile-was-betting-too-much-on-html5/

Mark Zuckerberg: Our Biggest Mistake Was Betting Too Much On HTML5

DREW OLANOFF

Tuesday, September 11th, 2012 **75 Comments**

Today, Mark Zuckerberg revealed that Facebook's mobile strategy relied too much on HTML5, rather than native applications.

Not only was this a big mistake with mobile, but Zuckerberg says that its biggest mistake period was the focus on HTML5. This is the first time that the Facebook CEO has openly admitted this, but things are looking good for the new iOS native app. According to Zuckerberg, people are consuming twice as many feed stories since the update to the new iOS app, which is great.

> *The first half year has been a little bit slow on product, but for the next six months I expect a lot of really cool stuff.*

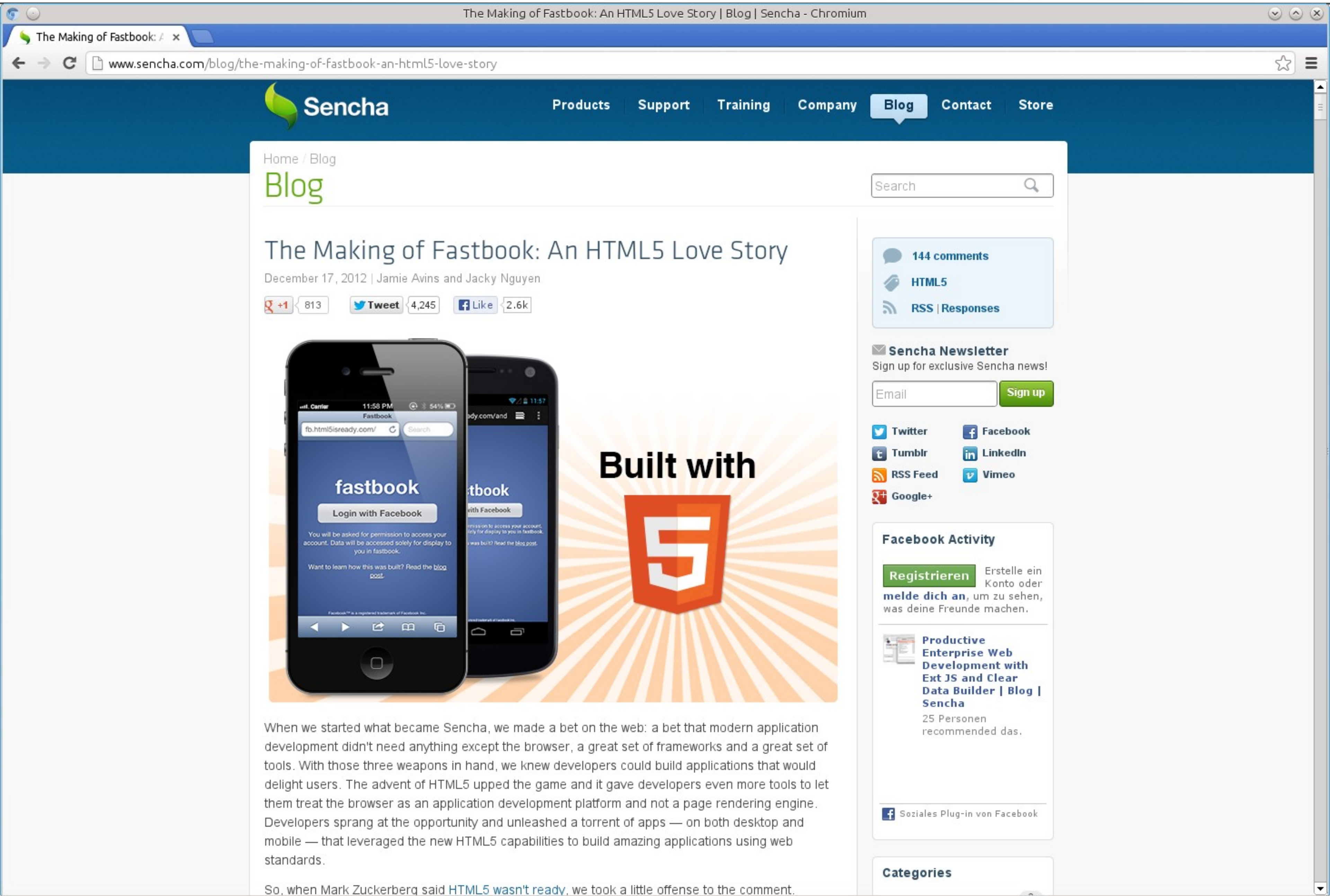
This "really cool stuff" will probably have monetization in mind, as it's very clear that **mobile is the path to ad revenue** for the company.

It's extremely difficult for a company to nose-dive into an adoption of a particular set of tools and then quickly change course. I suspect that this is exactly what happened with Facebook and things are at least looking up.

Last year, Facebook's CTO at the time, **Bret Taylor, discussed the future of mobile**, which at the time very much included HTML5. Here's a snippet from that piece, where Taylor discusses the platform:

> *Does that mean an evolution away from Flash? After all, Flash dominates the market for the types of HTML5 games that Facebook is talking about. "Well it's hard," Taylor said about Flash specifically. When I laughed and noted he was giving the diplomatic answer, he assured me that it is something they think about a lot. "We want to be ahead of the curve and fill in the gaps when possible," is how he ended up putting it.*

Offline strategies for HTML5 web applications



The Making of Fastbook: An HTML5 Love Story | Blog | Sencha - Chromium

www.sencha.com/blog/the-making-of-fastbook-an-html5-love-story

Sencha — Products · Support · Training · Company · Blog · Contact · Store

Home / Blog

Blog

The Making of Fastbook: An HTML5 Love Story

December 17, 2012 | Jamie Avins and Jacky Nguyen

+1 813 · Tweet 4,245 · Like 2.6k

When we started what became Sencha, we made a bet on the web: a bet that modern application development didn't need anything except the browser, a great set of frameworks and a great set of tools. With those three weapons in hand, we knew developers could build applications that would delight users. The advent of HTML5 upped the game and it gave developers even more tools to let them treat the browser as an application development platform and not a page rendering engine. Developers sprang at the opportunity and unleashed a torrent of apps — on both desktop and mobile — that leveraged the new HTML5 capabilities to build amazing applications using web standards.

So, when Mark Zuckerberg said HTML5 wasn't ready, we took a little offense to the comment.

144 comments
HTML5
RSS | Responses

Sencha Newsletter
Sign up for exclusive Sencha news!
Email · Sign up

Twitter · Facebook · Tumblr · LinkedIn · RSS Feed · Vimeo · Google+

Facebook Activity

Registrieren Erstelle ein Konto oder melde dich an, um zu sehen, was deine Freunde machen.

Productive Enterprise Web Development with Ext JS and Clear Data Builder | Blog | Sencha
25 Personen recommended das.

Soziales Plug-in von Facebook

Categories

What does „offline“ mean?

What does „offline“ mean?

Application vs. Content

What does „offline“ mean?

Application Cache vs. Offline Storage

App Cache for caching static resources

HTML Page:

```
<!DOCTYPE html>
<html lang="en">
```

App Cache for caching static resources

HTML Page:

```
<!DOCTYPE html>
<html lang="en" manifest="cache.manifest">
```

cache.manifest (served with Content-Type: text/cache-manifest):

```
CACHE MANIFEST

js/app.js
css/app.css
favicon.ico
http://someotherdomain.com/image.png
```

App Cache for caching static resources

```
CACHE MANIFEST
# 2013-07-25

NETWORK:
data.php

CACHE:
/main/home
/main/app.js
/settings/home
/settings/app.js
http://myhost/logo.png
http://myhost/check.png
http://myhost/cross.png
```

App Cache for caching static resources

```
CACHE MANIFEST
# 2013-07-25

FALLBACK:
/ /offline.html

NETWORK:
*
```

App Cache Scripting

```
// events fired by window.applicationCache
window.applicationCache.onchecking = function(e)
{log("Checking for updates");}
window.applicationCache.onnoupdate = function(e)
{log("No updates");}
window.applicationCache.onupdateready = function(e)
{log("Update ready");}
window.applicationCache.onobsolete = function(e)
{log("Obsolete");}
window.applicationCache.ondownloading = function(e)
{log("Downloading");}
window.applicationCache.oncached = function(e)
{log("Cached");}
window.applicationCache.onerror = function(e)
{log("Error");}

// Log each file
window.applicationCache.onprogress = function(e) {
  log("Progress: downloaded file " + counter);
  counter++;
};
```

App Cache Scripting

```
// Check if a new cache is available on page load.
window.addEventListener('load', function(e) {
  window.applicationCache.addEventListener('updateready',
  function(e) {

    if(window.applicationCache.status ==
        window.applicationCache.UPDATEREADY) {
      // Browser downloaded a new app cache.
      // Swap it in and reload the page
      window.applicationCache.swapCache();
      if (confirm('New version is available. Load it?)) {
        window.location.reload();
      }
    } else {
      // Manifest didn't change...
    }
  }, false);

}, false);
```

App Cache – Some gotchas!

App Cache – Some gotchas!

1. Files are always(!) served from the application cache.

App Cache – Some gotchas!

2. The application cache only updates if the content of the manifest itself has changed!

App Cache – Some gotchas!

3. If any of the files listed in the CACHE section can't be retrieved, the entire cache will be disregarded.

App Cache – Some gotchas!

4. If the manifest file itself can't be retrieved, the cache will ignored!

App Cache – Some gotchas!

5. Non-cached resources will not load on a cached page!

App Cache – Some gotchas!

6. The page needs to be reloaded, otherwise the new resources do not show up!

App Cache – Some gotchas!

7. To avoid the risk of caching manifest files set expires headers!

App Cache – What to cache?

Yes:
- Fonts
- Splash image
- App icon
- Entry page
- Fallback bootstrap

No:
- CSS
- HTML
- Javascript

App Cache – What to cache?

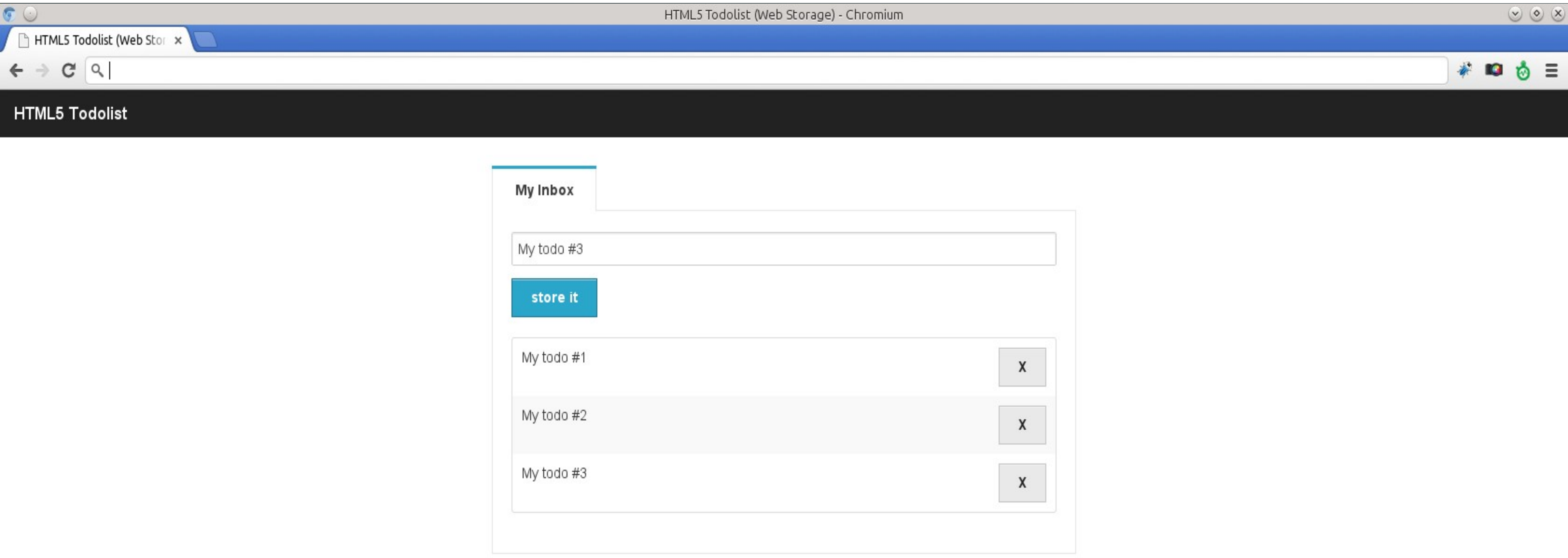
Use the app cache for „static content“ only!

Data URI Schema

Data URI Schema

```
<!DOCTYPE HTML>
<html>
 <head>
  <title>The Data URI scheme</title>
  <style type="text/css">
  ul.checklist li {
    margin-left: 20px;
    background: white
url('data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAAUAAA
AFCAYAAACNbyblAAAAHElEQVQI12P4//8/w38GIAXDIBKE0DHxgljNBAA
O9TXL0Y4OHwAAAABJRU5ErkJggg==') no-repeat scroll left
top;
}
  </style>
 </head>
 <body>
  <img
src="data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAAUAAA
AFCAYAAACNbyblAAAAHElEQVQI12P4//8/w38GIAXDIBKE0DHxgljNBAA
O9TXL0Y4OHwAAAABJRU5ErkJggg==" alt="Red dot">
 </body>
</html>
```

Storing dynamic data locally (in HTML5)



Example: Todolist application

Storing dynamic data locally (in HTML5)

Find the sources here:

github.com/bitExpert/html5-offline

Storing dynamic data locally (in HTML5)

Web Storage, Web SQL Database, IndexedDB, File API

Web Storage

Web Storage

Very convenient form of offline storage: simple key-value store

Web Storage: 2 different types

localStorage vs. sessionStorage

Web Storage: Add item

```
function add(item) {
    try {
        // for a new item set id
        if((typeof item.id === "undefined")
            || (null == item.id) || ("" == item.id)) {
            item.id = get_lastIndex() + 1;
        }

        // store object as string
        localStorage.setItem(item.id,
            JSON.stringify(item)
        );

        // update the index
        set_lastIndex(item.id);
    }
    catch(ex) {
        console.log(ex);
    }
}
```

Web Storage: Modify item

```
function modify(item) {
     try {
          // store object as string
          localStorage.setItem(item.id,
               JSON.stringify(item)
          );
     }
     catch(ex) {
          console.log(ex);
     }
}
```

Web Storage: Remove item

```
function remove (id) {
     try {
          localStorage.removeItem(id);
     }
     catch(ex) {
          console.log(ex);
     }
}
```

Web Storage: Read items

```
function read() {
      try {
          var lastIdx = get_lastIndex();
          for(var i = 1; i <= lastIdx; i++) {
              if(null !== localStorage.getItem(i)) {
                  // parse and render item
                  var item = JSON.parse(
                       localStorage.getItem(i)
                  );
              }
          }
      }
      catch(ex) {
          console.log(ex);
      }
}
```

Web Storage: How to use sessionStorage?

Web Storage: How to use sessionStorage?

Replace „localStorage“
with „sessionStorage“

Web Storage: Add item (sessionStorage style)

```
function add(item) {
     try {
          // for a new item set id
          if((typeof item.id === "undefined")
               || (null == item.id) || ("" == item.id)) {
               item.id = get_lastIndex() + 1;
          }

          // store object as string
          sessionStorage.setItem(item.id,
               JSON.stringify(item)
          );

          // update the index
          set_lastIndex(item.id);
     }
     catch(ex) {
          console.log(ex);
     }
}
```

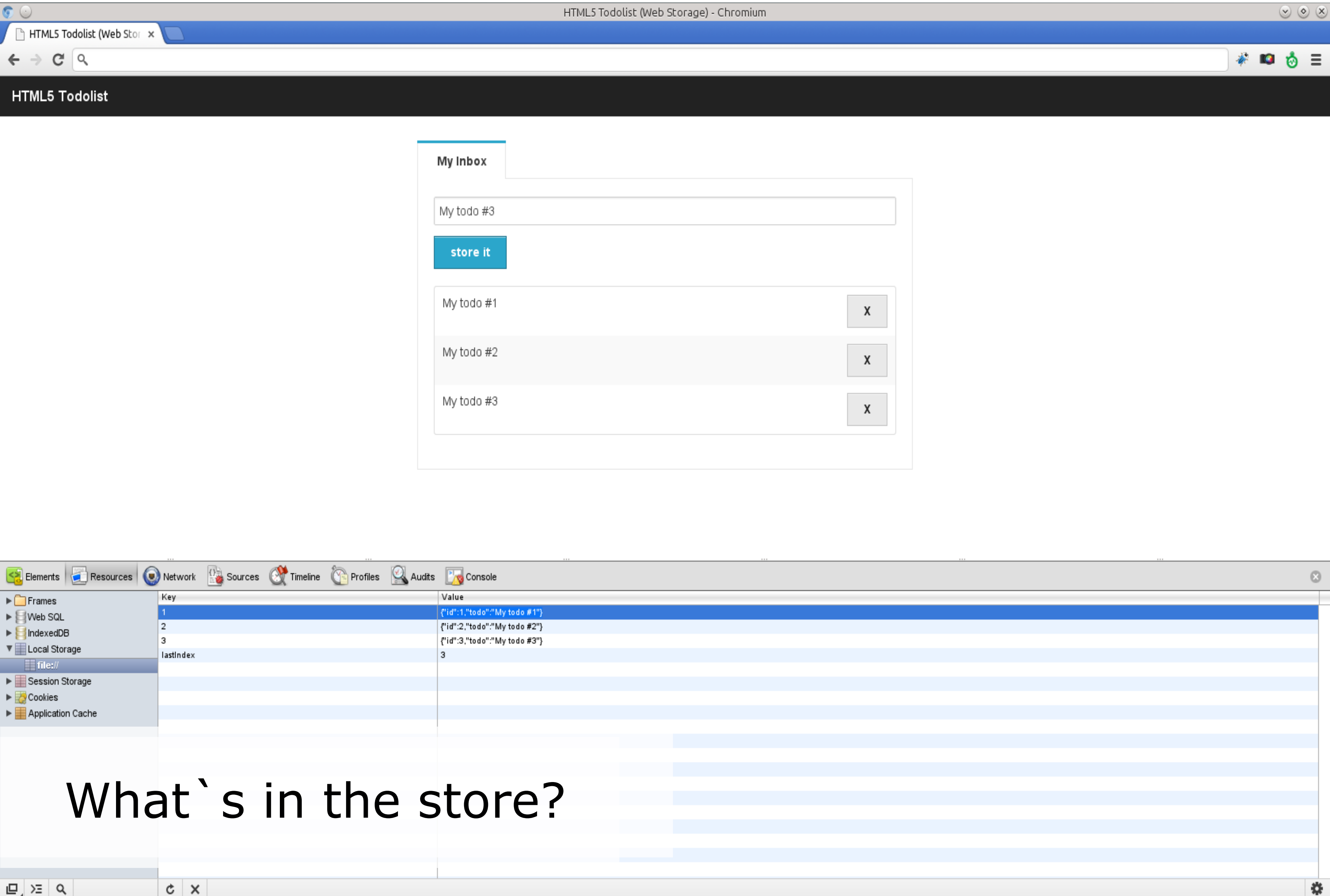
bitExpert

Web Storage: Don`t like method calls?

Web Storage: Don`t like method calls?

```
var value = "my value";

// method call
localStorage.setItem("key", value);

// Array accessor
localStorage[key] = value;

// Property accessor
localStorage.key = value;
```

Offline Strategien für HTML5 Web Applikationen



Key	Value
1	{"id":1,"todo":"My todo #1"}
2	{"id":2,"todo":"My todo #2"}
3	{"id":3,"todo":"My todo #3"}
lastIndex	3

What`s in the store?

Web Storage: Pro

Most compatible format up to now.

Web Storage: Con

The data is not structured.

Web Storage: Con

No transaction support!

Web Storage: Con

Lack of automatically expiring storage.

Web Storage: Con

Inadequate information about storage quota.

Web SQL Database

Web SQL Database

An offline SQL database based on SQLite, an general-purpose SQL engine.

Web SQL Database: Callback methods

```
var onError = function(tx, ex) {
    alert("Error: " + ex.message);
};

var onSuccess = function(tx, results) {
    var len = results.rows.length;

    for(var i = 0; i < len; i++) {
        // render found todo item
        render(results.rows.item(i));
    }
};
```

Web SQL Database: Setup Database

```
// initalize the database connection
var db = openDatabase('todo', '1.0', 'Todo Database',
   5 * 1024 * 1024 );

db.transaction(function (tx) {
     tx.executeSql(
          'CREATE TABLE IF NOT EXISTS todo '+
          '(id INTEGER PRIMARY KEY ASC, todo TEXT)',
          [],
          onSuccess,
          onError
     );
});
```

Web SQL Database: Add item

```
function add(item) {
    db.transaction(function(tx) {
        tx.executeSql(
            'INSERT INTO todo (todo) VALUES (?)',
            [
                item.todo
            ],
            onSuccess,
            onError
        );
    });
}
```

Web SQL Database: Modify item

```
function modify(item) {
    db.transaction(function(tx) {
        tx.executeSql(
            'UPDATE todo SET todo = ? WHERE id = ?',
            [
                item.todo
                item.id
            ],
            onSuccess,
            onError
        );
    });
}
```

Web SQL Database: Remove item

```
function remove(id) {
    db.transaction(function (tx) {
        tx.executeSql(
            'DELETE FROM todo WHERE id = ?',
            [
                id
            ],
            onSuccess,
            onError
        );
    });
}
```

Web SQL Database: Read items

```
function read() {
     db.transaction(function (tx) {
          tx.executeSql(
               'SELECT * FROM todo',
               [],
               onSuccess,
               onError
          );
     });
}
```

Web SQL Database: Pro

It`s a SQL database within the browser!

Web SQL Database: Con

It`s a SQL database within the browser!

Web SQL Database: Con

SQLite is slooooow!

Web SQL Database: Con

The specification is no longer part of HTML5!

IndexedDB

IndexedDB

A nice compromise between Web Storage and Web SQL Database giving you the best of both worlds.

IndexedDB: Preparation

```
// different browsers, different naming conventions
var indexedDB = window.indexedDB ||
   window.webkitIndexedDB || window.mozIndexedDB ||
   window.msIndexedDB;

var IDBTransaction = window.IDBTransaction ||
   window.webkitIDBTransaction;

var IDBKeyRange = window.IDBKeyRange ||
   window.webkitIDBKeyRange;
```

IndexedDB: Create object store

```
var db = null;
var request = indexedDB.open("todo");
request.onfailure = onError;
request.onsuccess = function(e) {
    db = request.result;
    var v = "1.0";
    if(v != db.version) {
        var verRequest = db.setVersion(v);
        verRequest.onfailure = onError;
        verRequest.onsuccess = function(e) {
            var store = db.createObjectStore(
                "todo",
                {
                    keyPath: "id",
                    autoIncrement: true
                }
            );
            e.target.transaction.oncomplete =
               function() {};
        };
    }
};
```

bitExpert

IndexedDB: Add item

```
function add(item) {
     try {
          var trans = db.transaction(["todo"],
               IDBTransaction.READ_WRITE);

          var store   = trans.objectStore("todo");
          var request = store.put({
               "todo": item.todo,
          });
     }
     catch(ex) {
          onError(ex);
     }
}
```

IndexedDB: Modify item

```
function modify(item) {
     try {
          var trans = db.transaction(["todo"],
               IDBTransaction.READ_WRITE);

          var store   = trans.objectStore("todo");
          var request = store.put(item);
     }
     catch(ex) {
          onError(ex);
     }
}
```

IndexedDB: Remove item

```
function remove(id) {
     try {
          var trans = db.transaction(["todo"],
               IDBTransaction.READ_WRITE);

          var store   = trans.objectStore("todo");
          var request = store.delete(id);
     }
     catch(ex) {
          onError(ex);
     }
}
```

IndexedDB: Read items

```
function read () {
    try {
        var trans = db.transaction(["todo"],
            IDBTransaction.READ);

        var store = trans.objectStore("todo");
        var keyRange = IDBKeyRange.lowerBound(0);
        var cursorRequest = store.openCursor(keyRange);

        cursorRequest.onsuccess = function(e) {
            var result = e.target.result;
            if(!!result == false) {
                return;
            }
            // @TODO: render result.value
            result.continue();
        };
    }
    catch(ex) {
        onError(ex);
    }
}
```

bitExpert

File API

File API

FileReader API and FileWriter API

File API: Preparations

```
var onError = function(e) {
    var msg = '';

    switch(e.code) {
        case FileError.QUOTA_EXCEEDED_ERR:
            msg = 'QUOTA_EXCEEDED_ERR'; break;
        case FileError.NOT_FOUND_ERR:
            msg = 'NOT_FOUND_ERR'; break;
        case FileError.SECURITY_ERR:
            msg = 'SECURITY_ERR'; break;
        case FileError.INVALID_MODIFICATION_ERR:
            msg = 'INVALID_MODIFICATION_ERR'; break;
        case FileError.INVALID_STATE_ERR:
            msg = 'INVALID_STATE_ERR'; break;
        default:
            msg = 'Unknown Error'; break;
    };

    alert("Error: " + msg);
};
```

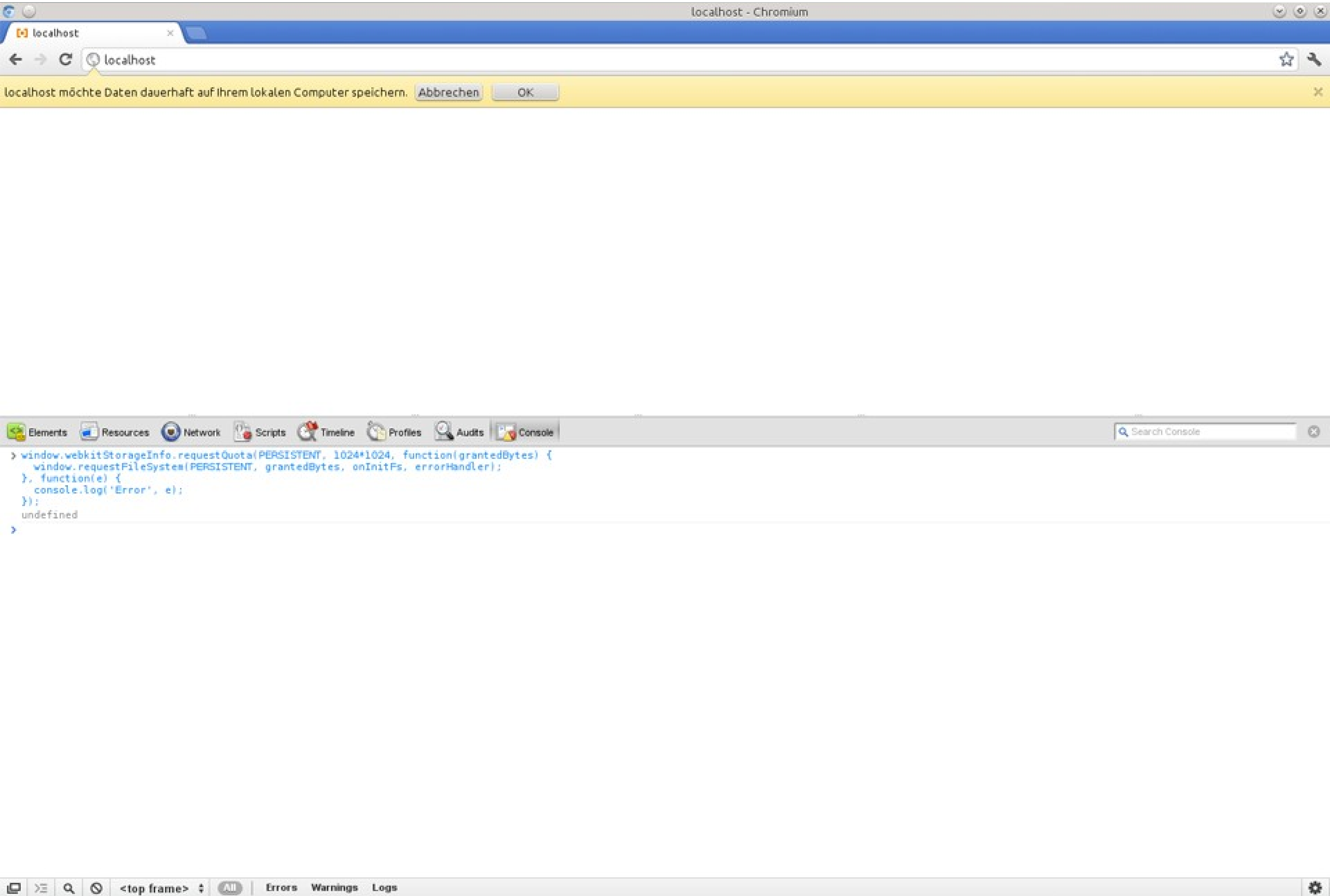
File API: Preparations

```
// File system has been prefixed as of Google Chrome 12
window.requestFileSystem = window.requestFileSystem ||
    window.webkitRequestFileSystem;

window.BlobBuilder = window.BlobBuilder ||
    window.WebKitBlobBuilder;

var size = 5 * 1024*1024; // 5MB
```

File API: Requesting quota

```
// request quota for persistent store
window.webkitStorageInfo.requestQuota(
     PERSISTENT,
     size,
     function(grantedBytes) {
          window.requestFileSystem(
               PERSISTENT,
               grantedBytes,
               function(fs) {
                    // @TODO: access filesystem
               }
          }
     }
}
```

Offline strategies for HTML5 web applications


localhost - Chromium

localhost möchte Daten dauerhaft auf Ihrem lokalen Computer speichern.

```
> window.webkitStorageInfo.requestQuota(PERSISTENT, 1024*1024, function(grantedBytes) {
    window.requestFileSystem(PERSISTENT, grantedBytes, onInitFs, errorHandler);
  }, function(e) {
    console.log('Error', e);
  });
  undefined
>
```

File API: Add item

```
function add(item) {
      window.webkitStorageInfo.requestQuota(
          PERSISTENT,
          size,
          function(grantedBytes) {
              window.requestFileSystem(
                  PERSISTENT,
                  grantedBytes,
                  function(fs){
                      writeToFile(fs, item);
                  },
                  onError
              );
          },
          function(e) {
              onError(e);
          }
      );
  },
```

File API: Add item

```
function writeToFile(fs, item) {
    fs.root.getFile(
        'todo.txt',
        {
            create: true
        },
        function(fileEntry) {
            fileEntry.createWriter(
                function(fileWriter) {
                    var bb = new window.BlobBuilder();
                    bb.append(JSON.stringify(item)+
                        "\n");

                    fileWriter.seek(fileWriter.length);
                    fileWriter.write(
                        bb.getBlob('text/plain'));
                }, onError
            );
        }, onError
    );
};
```

File API: Add item

```
function writeToFile(fs, item) {
    fs.root.getFile(
        'todo.txt',
        {
            create: true
        },
        function(fileEntry) {
            fileEntry.createWriter(
                function(fileWriter) {
                    var bb = new window.BlobBuilder();
                    bb.append(JSON.stringify(item)+
                        "\n");

                    fileWriter.seek(fileWriter.length);
                    fileWriter.write(
                        bb.getBlob('text/plain'));
                }, onError
            );
        }, onError
    );
};
```

Deprecated!

File API: Add item

```
function writeToFile(fs, item) {
    fs.root.getFile(
        'todo.txt',
        {
            create: true
        },
        function(fileEntry) {
            fileEntry.createWriter(
                function(fileWriter) {
                    var blob = new Blob([
                        JSON.stringify(item)+"\n"
                    ]);

                    fileWriter.seek(fileWriter.length);
                    fileWriter.write(blob);
                }, onError
            );
        }, onError
    );
};
```

File API: Read items

```
function read() {
        window.webkitStorageInfo.requestQuota(
             PERSISTENT,
             size,
             function(grantedBytes) {
                  window.requestFileSystem(
                       PERSISTENT,
                       grantedBytes,
                       function(fs){
                            readFromFile(fs);
                       },
                       onError
                  );
             },
             function(e) {
                  onError(e);
             }
        );
}
```

File API: Read items

```
function readFromFile(fs) {
     fs.root.getFile(
          'todo.txt',
          {
               create: true
          },
          function(fileEntry) {
               fileEntry.file(function(file){
                    var reader = new FileReader();
                    reader.onloadend = function(e) {
                         if (evt.target.readyState ==
                              FileReader.DONE) {
                              // process this.result
                         }
                    };
                    reader.readAsText(file);
               });
          }, onError
     );
}
```

Am I online?

Am I online?

```
document.body.addEventListener("online", function () {
  // browser is online!
}

document.body.addEventListener("offline", function () {
  // browser is not online!
}
```

Am I online? Another approach...

```
$.ajax({
  dataType: 'json',
  url: 'http://myappurl.com/ping',
  success: function(data){
    // ping worked
  },
  error: function() {
    // ping failed -> Server not reachable
  }
});
```

How to sync your data?

How to sync your data?

PouchDB, the JavaScript Database that syncs!

How to sync your data?

```
var db = new PouchDB('todo');

db.put({
 _id: 1,
 todo: 'Get some work done...',
});

db.replicate.to('http://example.com/mydb');
```

noBackend solution

noBackend solution

An approach to decouple apps from backends, by abstracting backend tasks with frontend code.

noBackend solution

„Hoodie is an architecture for frontend-only web apps"

Browser support?

Browser support?

	App Cache	Web Storage	WebSQL	IndexedDB	File API
IE	10.0	8.0	10.0	10.0	-
Firefox	11.0	11.0	11.0	11.0	19.0
Chrome	18.0	18.0	18.0	18.0	18.0
Safari	5.1	5.1	5.1	-	-
Opera	12.1	12.1	12.1	-	-
iOS Safari	3.2	3.2	3.2	-	-
Android	2.1	2.1	2.1	-	-

Source: http://caniuse.com

Storage limitations?

Storage limitations?

All storage technologies are limited by quotas. Be aware of what you do!

Storage limitations?

	App Cache	Web Storage	WebSQL	IndexedDB	File API
iOS 5.1	10 MB	5 MB	5 MB	5 MB	
Android 4	unlimited	5 MB	?	?	
Safari 5.2	unlimited	5 MB	5 MB	5 MB	
Chrome 18	5 MB	5 MB	unlimited	unlimited	unlimited
IE 10	50 MB	10 MB	500 MB	500 MB	
Opera 11	50 MB	5 MB	5 MB	5 MB	
Firefox 11	unlimited	10 MB	50 MB	50 MB	

bitExpert

Thank you!