

Offline Apps mit HTML5

Entwicklung vollständig offline-tauglicher Webanwendungen

W3L AG
info@W3L.de

2014



Agenda

■ Motivation

- Situation unter HTML 4.01
- Neuerungen durch HTML5

■ Technologie

- Offline-Technologien
- Praktischer Einsatz
- jStorage

■ HTML5 Offline Web App

- Beispielanwendung

■ Fazit



Situation unter HTML 4.01

- **Webapps benötigen eine stetige Verbindung zum Server**
 - Verbindungsabbrüche führen zu Verlust ungespeicherter Eingaben
 - Keine weitere Arbeit mit der Web-Anwendung nach Abbruch möglich
 - Inkonsistenter Zustand nach Wiederherstellung der Verbindung

- **Besondere Relevanz für Webapps, in denen WWW-Anbindung nicht der zentrale Punkt der Applikation ist**
 - Grafikeditoren
 - Textverarbeitungen
 - Kalender, To-Do-Listen
 - etc.

Historie: userData Behavior

- **Stark veraltet: Mit Internet Explorer 4 eingeführt**
 - Nur verfügbar von Internet Explorer 4 bis 7 und auch nur dort ratsam
- **Daten werden nicht in globales DOM-Objekt abgelegt, sondern in Elementen auf der Seite**

```
<style>
  .speicherbar {
    behavior:url(#default#userData);
  }
</style>
```

```
<script>
  var saveText = document.getElementById('saveText');
  function saveInput() {
    saveText.setAttribute('datenspeicher', saveText.value);
    saveText.save('meinDatenspeicher');
  }
  function restoreInput() {
    saveText.load('meinDatenspeicher');
    saveText.value = saveText.getAttribute('datenspeicher');
  }
</script>
```

Neuerungen durch HTML5

■ Webapps können nun vollständig offline-tauglich sein

- Gezieltes Vorladen und Cachen von Ressourcen
- Datenbankeinträge zwischenspeichern für späteren Versand
- Bei konsequenter Gestaltung bemerkt der Anwender nicht, ob er gerade online ist oder nicht.

■ Anforderungen für den Offlinebetrieb der Webanwendung

- Erkennung, ob der Browser off- oder online ist
 - On- und Offline-Events im navigator-Objekt
- Speicherung statischer Ressourcen
 - Application Cache
- Form einer clientseitigen Datenbank
 - Siehe nachfolgende Folie

Offline-Technologien

■ Web Storage (auch DOM Storage)

- Weiterentwicklung von Cookies („Super Cookies“)
- Speicherung: Lokal (`localStorage`) und Session (`sessionStorage`)
- Am wenigsten leistungsstark, jedoch höchste Verbreitung (kein Opera Mini)

■ Web SQL Storage

- Keine Unterstützung für Internet Explorer (Mobile), Opera Mini und Firefox
- Im November 2010 eingestellt

■ Indexed Database API / Indexed DB

- Von Oracle entwickelte Alternative zu Web SQL
- NoSQL-Konzept
- Keine Unterstützung für (iOS) Safari und Opera Mini

Offline-Technologien

■ HTML5 Application Cache

- Mechanismus zum Speichern von Ressourcen für den Offline-Betrieb
- Nicht zu verwechseln mit Browsercache
- Präzise steuerbar durch den Webentwickler
- Was gespeichert wird, wird im Cache Manifest festgelegt
- Manifest-Datei ist eine normale Textdatei
 - MIME-Type: `text/cache-manifest`
 - Gewählte Endung muss in `.htaccess` eingetragen werden
 - Für `.manifest: AppType text/cache-manifest manifest`

CACHE MANIFEST

Die erste Zeile ist Pflicht. Dies ist ein Kommentar

```
img/foo.png  
scripts/bar.js
```

```
FALLBACK:  
/ offline.html
```

Offline-Technologien

■ Desktop Browser

Technologie	Firefox	Safari	Chrome	Opera	Internet Explorer
Application Cache	3.5+	4.0+	4.0+	10.60+	10.0+
Web Storage	3.5+	4.0+	1.0+	10.50+	8.0+
Web SQL	-	4.0+	4.0+	10.50+	-
Indexed DB	4.0+ *	-	11.0+	-	10.0+ **
On/Offline-Events	- ***	5.0.4+	11.0+	- ***	8.0+ ****
On/Offline-Status	✓	✓	✓	✓	4.0+

- * Implementierung unvollständig
- ** Experimentelle Implementierung für IE 9
- *** Fehlerhafte Implementierung – tatsächlicher Status wird nicht beachtet
- **** Abweichende Umsetzung im IE 8

Offline-Technologien

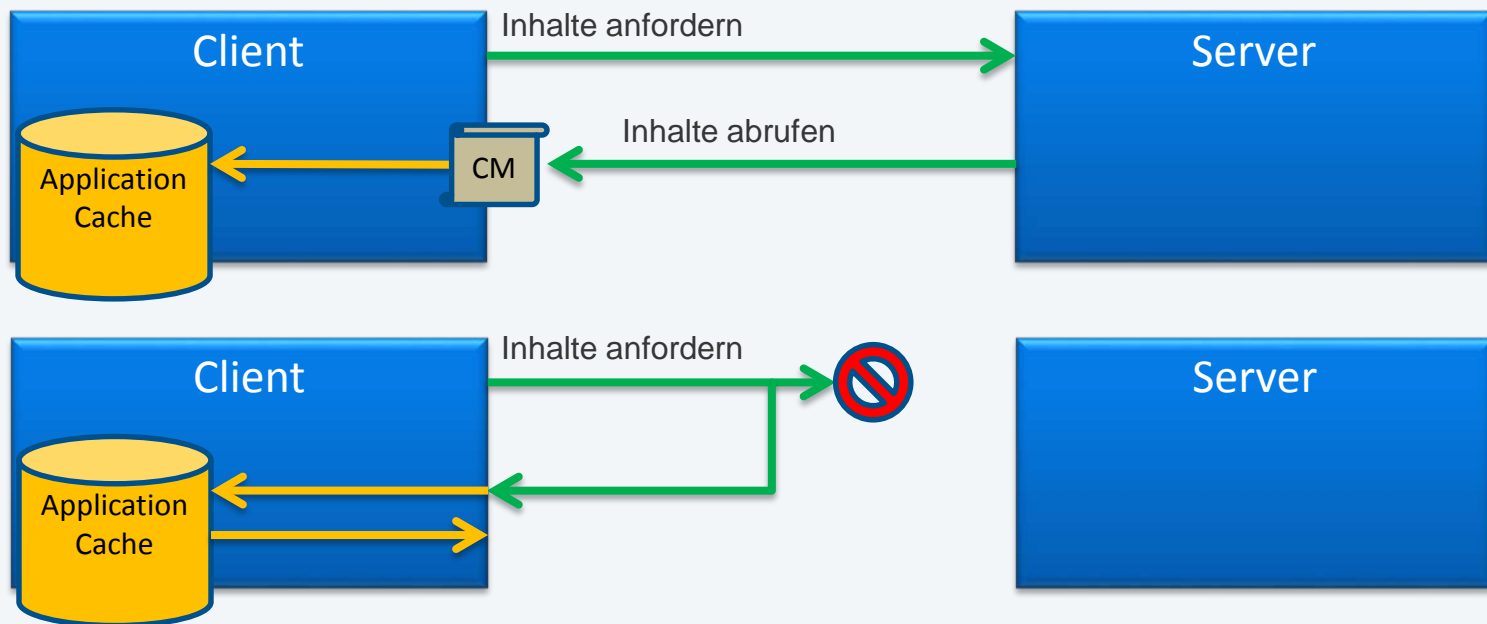
■ Mobile Browser

Technologie	iOS Safari	Opera Mobile	Android Browser	Blackberry Browser	IE Mobile
Application Cache	3.2+	✓*	2.1+	7.0+	10.0+
Web Storage	3.2+	✓*	2.1+	7.0+	10.0+
Web SQL	3.2+	✓*	2.1+	7.0+	-
Indexed DB	-	✓*	4.4+	10.0+	10.0+
On/Offline-Events	3.2+	✓*	2.1+	7.0+	10.0+
On/Offline-Status	✓	✓*	✓	✓	✓

- * Keine einheitliche Versionisierung
 - Android: v.16
 - Windows Mobile: v.10.0
 - etc.

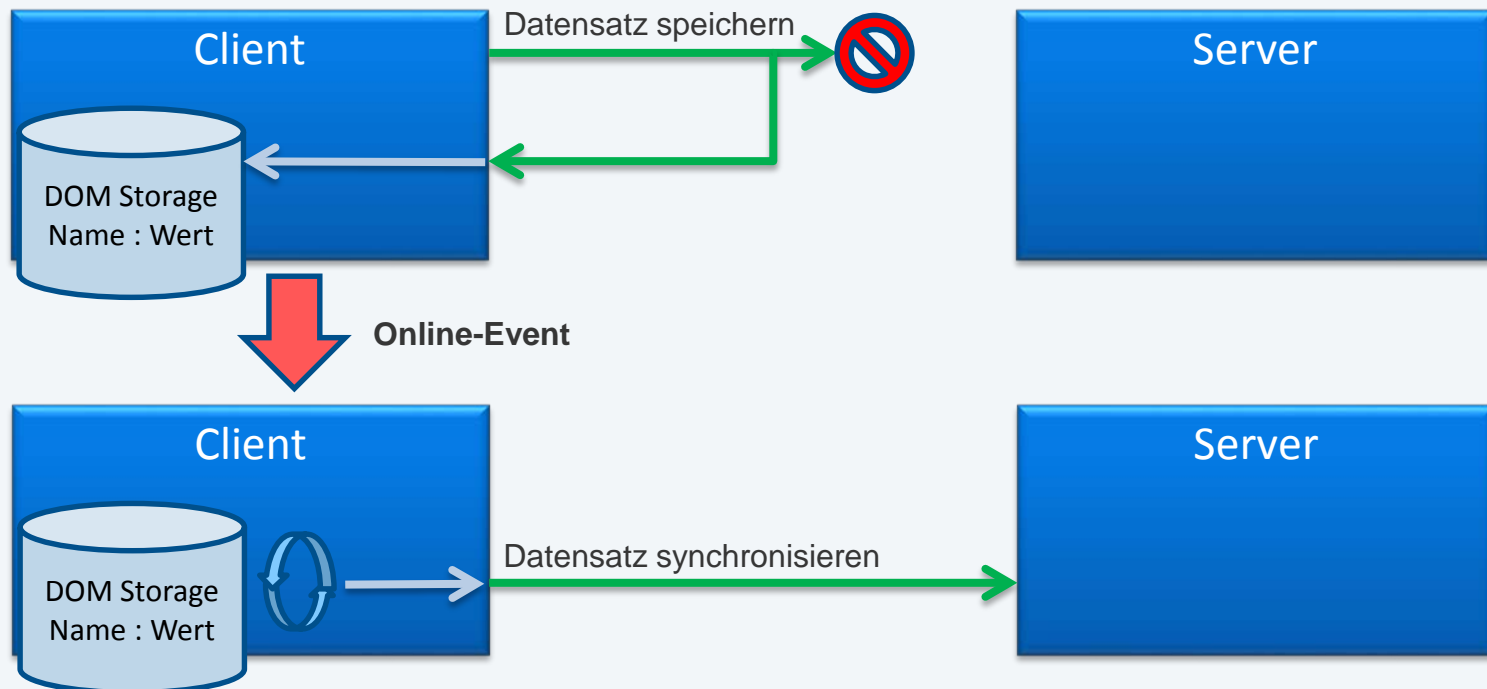
Praxis: Application Cache

- Client ruft Inhalte vom Webserver ab.
- Cache Manifest legt fest, ob und welche Inhalte (HTML-Seiten, Bilder, Skripts) im Application Cache gespeichert werden.
- Ist der Server beim nächsten Aufruf offline, können Inhalte aus dem Application Cache abgerufen werden.



Praxis: Web Storage

- Szenario: Client möchte Datensätze auf dem Server speichern. Die Verbindung zum Server geht allerdings verloren.
- Client speichern die Datensätze zunächst lokal im DOM Storage.
- Client erhält die Mitteilung via Event, dass die Verbindung wiederhergestellt wurde und synchronisiert die lokalen Datensätze mit den Serverdaten.



Praxis: Web SQL Storage

■ Im Vergleich zu DOM Storage wesentlicher leistungsstärker

- SQL-Syntax und Speicherung komplexerer Daten

■ Eine Datenbankverbindung aufbauen

```
var name = "database";
var version = "1.0";
var title = "Testdatenbank";
var max = 81920;
var db = openDatabase(name, version, title, max);
```

■ Queries ausführen wenn die Verbindung steht

```
if(typeof db !== 'undefined') {
  execute = document.getElementById('execute');
  execute.addEventListener('click', function() {
    if(query.value) {
      db.transaction(function(t) {
        t.executeSql(query.value);
      });
    }
  });
};
```

Praxis: Indexed DB

- Zukünftiger Standard für die Speicherung von Offline-Daten
- Namespace mit Datenbanklogik

```
var html5rocks = {};  
html5rocks.indexedDB = {};
```

- Datenverbindung wird aufgebaut, nach Erfolg können weitere Schritte folgen

```
html5rocks.indexedDB.db = null;  
  
html5rocks.indexedDB.open = function() {  
  var request = indexedDB.open("database");  
  
  request.onsuccess = function(e) {  
    html5rocks.indexedDB.db = e.target.result;  
    // Logik folgt hier  
  };  
  
  request.onfailure = html5rocks.indexedDB.onerror;  
};
```

jStorage

- Plugin für jQuery, MooTools und Prototype
- Komfortable Vereinfachung der Offline-Speicherung
- Für alle Browser ab Internet Explorer 6

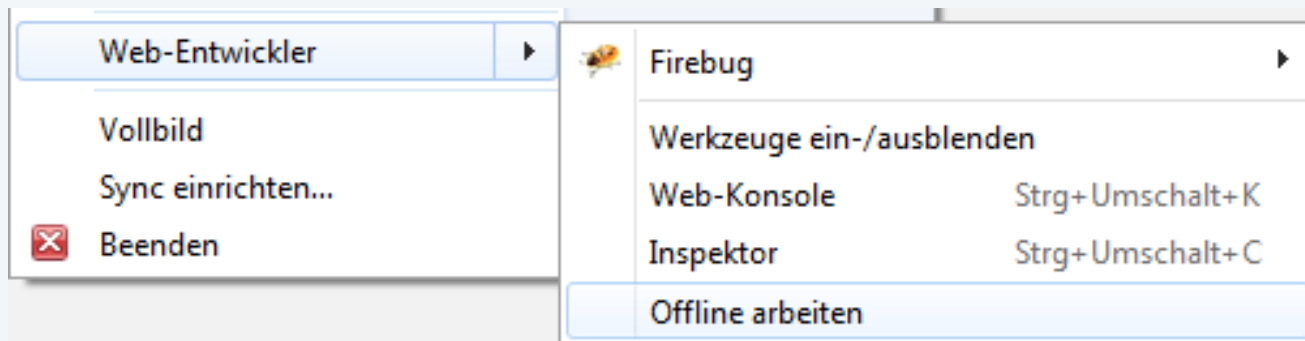
```
// Können wir Daten speichern?  
if($.jStorage.storageAvailable()) {  
  
    // Datensatz "bar" an Key "foo" speichern  
    $.jStorage.set('foo', 'bar');  
  
    // Datensatz vom Key "foo" holen, "Standardwert" zurückgeben,  
    // falls nicht vorhanden  
    var val = $.jStorage.get('foo', "Standardwert");  
}
```

Umsetzung von Offline-Technologien anhand eines Beispiels

HTML5 OFFLINE WEB APP

Beispielanwendung

- Einfache ToDo-Liste, die Einträge speichert und ausgibt
- Demonstration
 - Speicherung von Ressourcen im Application Cache
 - Speicherung von Datensätzen mittels DOM Storage
- Offline-Zustand wird via Entwickler-Option „Offline-Modus“ im Browser simuliert
 - Die Web App muss diesen Status des Browsers berücksichtigen



Fazit

■ Stärken Offline Web Apps

- Verständliche, leicht erlernbare Implementierung
- Ungestörte Nutzung der Anwendung bei Verbindungsverlust
- Ältere Browser können die Technik teilweise nutzen
 - Internet Explorer 4.0 bis 7.0: userData Behavior

■ Probleme von Offline Web Apps

- Nur Web Storage auf nahezu allen Browsern zugleich nutzbar
 - z.B. Indexed DB unterstützt noch nicht alle Browser
 - Web SQL Storage ist bereits eingestellt
- Implementierung bei Browsern uneinheitlich
 - On- und Offline-Status werden teils unterschiedlich interpretiert

Literatur

- „HTML5 – Webseiten innovativ und zukunftssicher“ von Peter Kröner
- http://www.w3schools.com/html/html5_app_cache.asp
- <http://www.html5rocks.com/de/tutorials/indexeddb/todo/>
- <https://developer.mozilla.org/en-US/docs/Web/API/NavigatorOnline>
- <http://diveintohtml5.info>
- <http://www.jstorage.info>
- <http://caniuse.com/>

Vielen Dank!

Inhouse-Schulungen



Wir bieten Inhouse-Schulungen und Beratung durch unsere IT-Experten und -Berater.

Schulungsthemen

- Softwarearchitektur (OOD)
- Requirements Engineering (OOA)
- Nebenläufige & verteilte Programmierung

Gerne konzipieren wir auch eine individuelle Schulung zu Ihren Fragestellungen.



Sprechen Sie uns an!
Tel. 0231/61 804-0, info@W3L.de

W3L-Akademie



Flexibel online lernen und studieren!

In Zusammenarbeit mit der Fachhochschule Dortmund bieten wir

zwei Online-Studiengänge

- B.Sc. Web- und Medieninformatik
- B.Sc. Wirtschaftsinformatik

und 7 Weiterbildungen im IT-Bereich an.



Besuchen Sie unsere Akademie!
<http://Akademie.W3L.de>