

# HTML5 Web Storage

Ingo Köster

Diplom-Informatiker (FH)

# Storage

---

## › Unterschiede zu Cookies

- › Mittels Storage können nicht nur wenige Kilobytes sondern mehrere Megabytes gespeichert werden
- › Gespeicherte Daten werden nicht zwischen Server und Client hin und hergeschickt
- › Storage Daten laufen nicht ab (bei `localStorage`)

## › Gemeinsamkeiten mit Cookies

- › Daten sind nicht vorhanden, wenn Benutzer andere Browser oder Computer verwenden
  - › Daten müssten dann doppelt gesichert werden

# Storage

---

- › Speichern von Werten als Name-Wert-Paar
- › Daten liegen unverschlüsselt auf der Festplatte
- › Zwei verschiedene Storage Klassen
- › `sessionStorage`
  - › Speichert nur so lange, wie die Session existiert
    - › Kann länger sein, falls der Browser das Fortsetzen einer Session nach einem Neustart unterstützt
- › `localStorage`
  - › Speichert dauerhaft (bis z.B. Cache geleert wird)

# Same-Origin-Policy (SOP)

---

- › Skripte einer Webseite können im Allgemeinen nur auf die Daten zugreifen, die diese selbst erstellt haben
- › Ausschlaggebend ist die Quelle
- › Webserver als Quelle setzt sich zusammen aus
  - › Protokoll (http oder https)
  - › Hostname
  - › Port

# Same-Origin-Policy

---

- › Ein Skript in `http://www.beispiel.de/verz/seite1.html` versucht auf Elemente in anderen Seiten zuzugreifen:

<code>http://www.beispiel.de/verz/seite2.html</code>	Ja
<code>http://www.beispiel.de/verz2/seite3.html</code>	Ja
<code>http<sup>s</sup>://www.beispiel.de/x.html</code>	Nein
<code>http://<sup>shop</sup>.beispiel.de/x.html</code>	Nein
<code>http://<sup>beispiel</sup>.de/x.html</code>	Nein
<code>http://www.beispiel.de:<sup>8080</sup>/x.html</code>	Nein

# Same-Origin-Policy

---

- › `sessionStorage` ist auch auf Fenster bzw. Tab beschränkt
- › D.h. gleiche Quelle in zwei Fenstern/Tabs geöffnet bedeutet:
- › 2 mal `sessionStorage`, welche nicht gegenseitig auf die Daten des Anderen zugreifen können

# Web Storage

---

- › Web Storage wird von allen gängigen Browsern unterstützt
- › Der verfügbare Platz zum Speichern von Daten beträgt pro Same-Origin in der Regel mindestens 5 MB
  - › Kann je nach Browser mehr sein

# Daten speichern

---

- › Werte werden mit `localStorage.setItem` gespeichert
  - › bzw. `sessionStorage.setItem`
- › Für jeden gespeicherten Wert wird ein Schlüssel vergeben
  - › `localStorage.setItem("key", "wert");`
- › Falls der Schlüssel noch nicht existiert, wird dieser angelegt
- › Ansonsten wird der Wert überschrieben
- › Das Schreiben kann Exceptions auslösen



# Daten laden

---

- › Werte mit `localStorage.getItem` lesen
  - › bzw. `sessionStorage.getItem`
- › `localStorage.getItem("key");`
- › Ist der Schlüssel nicht vorhanden, wird `null` zurückgeliefert

# Alternativer Abruf der Daten

---

- › Über `localStorage.length` lässt sich die Anzahl der gespeicherten Werte abfragen
- › Web Storage verhält sich wie assoziatives Array
- › Iterieren über die Daten z.B. mittels for-Schleife
  - › `for (let value in localStorage) { ... }`
- › Daher auch folgende Syntax möglich
  - › `let derwert = localStorage["key"];`
  - › `let derwert = localStorage.key;`

# Daten löschen

---

- › Mittels `removeItem` einzelne Werte aus dem Storage löschen

```
localStorage.removeItem("key");
```

- › `removeItem` hat keinen Rückgabewert
- › Löst keinen Fehler aus, falls der Schlüssel nicht existiert

- › Mittels `clear` alle gespeicherten Daten einer Quelle löschen

```
localStorage.clear();
```

- › Löst keinen Fehler aus, falls Storage bereits leer ist

# Objekte speichern

---

- › Storage unterstützt nur Key-Value-Paare
- › Datenstrukturen wie Objekte müssen serialisiert bzw. deserialisiert werden, um diese Speichern und Laden zu können
- › Dazu können die Methoden `JSON.stringify` und `JSON.parse` verwendet werden
  - › Unterstützt keine zirkulären Referenzen

# Storage einsehen

---

- › Chrome: Mit Hilfe der Entwicklertools das Storage unter dem Punkt Application/Storage einsehen
- › Opera & Edge: Wie Chrome
- › Firefox: Werkzeugkasten-Einstellungen (Zahnrad) -> „Speicher“ aktivieren

