Taking the Web Offline

Erik Runyon
@erunyon

#AIM10
October 2014

<about/>

Erik Runyon

Director of Web Communications

The University of Notre Dame

@erunyon

weedygarden.net

#AIM10

Today's Topics

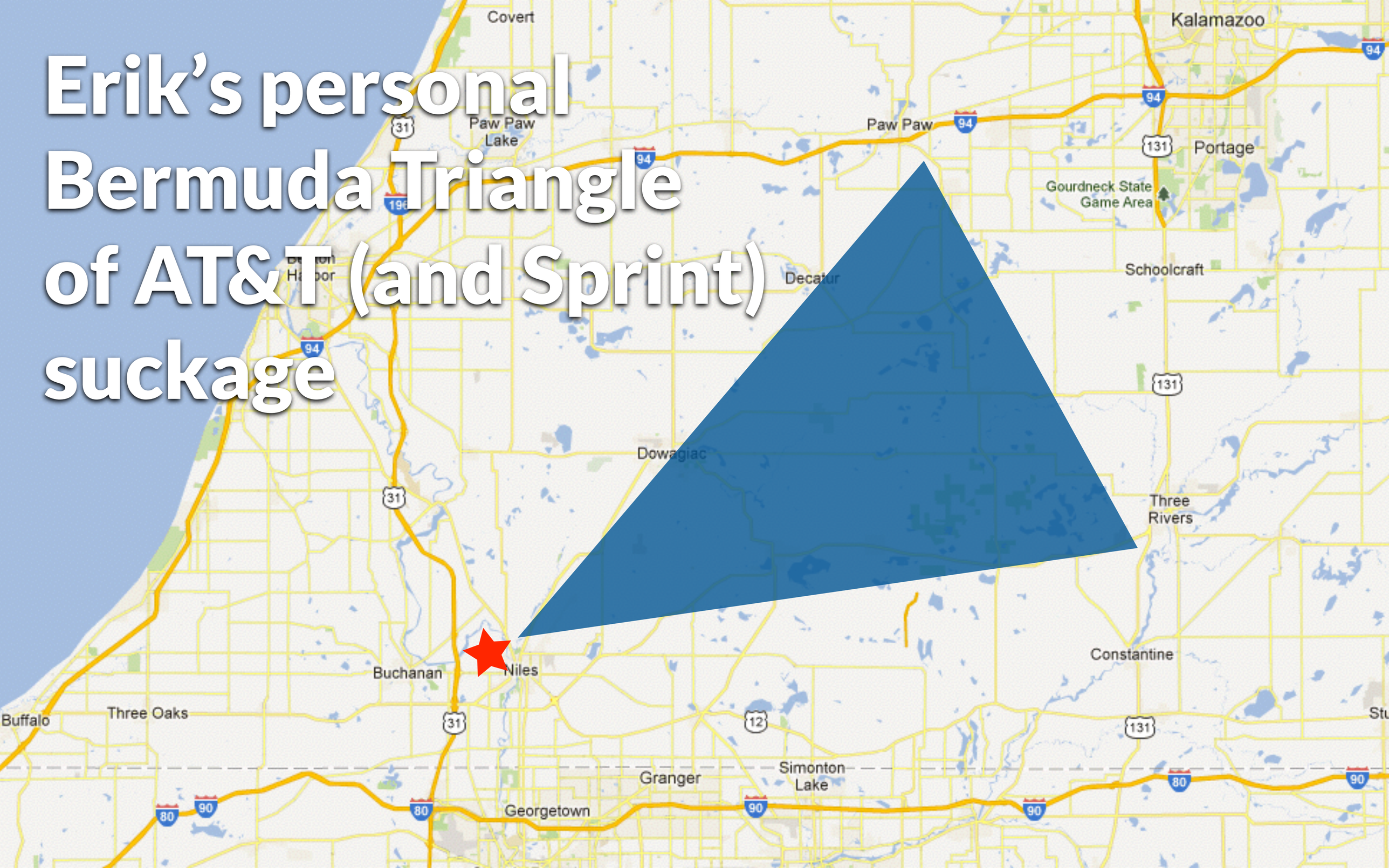
I. Why?

II. The Past

III. The Future

I. Why?

Erik's personal Bermuda Triangle of AT&T (and Sprint) suckage



Service Unavailable

* not a true story

Service REALLY Unavailable

Speed!!!!

We want an app…

II. The Past

Cookies

- Introduced in 1994 by the Netscape Communications Corporation
- Data passed to the server
- Store up to 4KB per cookie

Less notable solutions

Flash “Cookies” (Local Shared Objects) 100KB

Google Gears (deprecated in 2011)

III. The Future

Where we're going, we don't need… reliable internet connections.

Web SQL Database

Wrapper around SQLite

SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine.

http://www.w3.org/TR/webdatabase/

Web SQL Support

Chrome	Android	Safari	iOS	Firefox	IE
4+	2.1+	3.1+	3.2+	None	None*

* Not currently planned (status.modern.ie)

The nail in the coffin…

Spec is no longer in active maintenance.

Beware. This specification is no longer in active maintenance and the Web Applications Working Group does not intend to maintain it further.

IndexedDB

What it is…

IndexedDB is a client side storage API that persists data in a user's browser. It is a transactional, non-relational storage mechanism that saves key-value pairs in Object Stores and allows searching data using indexes.

What it’s not…

IndexedDB is NOT a Relational Database

http://docs.webplatform.org/wiki/apis/indexedDB

IndexedDB Support

Chrome	Android	Safari	iOS	Firefox	IE
					
23+	4.4+	8	8	10+	10+*

* a number of sub features are not supported in IE 10 & 11

Opening

```
// Opening a Database
var db;
var request = window.indexedDB.open("HEWeb 2014 Presentations", 1);

request.onerror = function(event) {
  console.log("Database error: " + event.target.errorCode);
};
request.onupgradeneeded   = function(event){
  console.log("Upgrading");
  db = event.target.result;
  var objectStore = db.createObjectStore("presentations", { keyPath : "uniq_id" });
};
request.onsuccess = function(event){
  console.log("Success opening DB");
  db = event.target.result;
};
```

Adding

```
// Adding an object
var uniq_id = "4cb507e9-56c5-478b-9ba6-87f4e509db38";
var title = "Taking the web offline";
var transaction = db.transaction(["presentations"],"readwrite");

var objectStore = transaction.objectStore("presentations");
objectStore.add({uniq_id: uniq_id, title: title});
```

Deleting

```
var uniq_id = "4cb507e9-56c5-478b-9ba6-87f4e509db38";

// Removing Object from ObjectStore
var transaction = db.transaction(["presentations"],"readwrite");
transaction.objectStore("presentations").delete(uniq_id);
```

Accessing an object

```
var uniq_id = "4cb507e9-56c5-478b-9ba6-87f4e509db38";

// Accessing an object with the key
var transaction = db.transaction(["presentations"],"readwrite");
var request = transaction.objectStore("presentations").get(uniq_id);

request.onsuccess = function(event){
  console.log("Title : " + request.result.title);
};
```

Updating

```
// Updating an Object
var uniq_id = "4cb507e9-56c5-478b-9ba6-87f4e509db38";
var transaction = db.transaction(["presentations"],"readwrite");
var objectStore = transaction.objectStore("presentations").get(uniq_id);
var request = objectStore.get(uniq_id);

request.onsuccess = function(event){
  request.result.title = "Taking the web… 😎 offline. YEAHHHHH!!!!";
  objectStore.put(request.result);
};
```

Combining IndexedDB and Web SQL?

- git.yathit.com/ydn-db/wiki/Home
- github.com/axemclion/IndexedDBShim
- github.com/mozilla/localForage

IndexedDB + Web SQL

Chrome	Android	Safari	iOS	Firefox	IE
4+	2.1+	3.1+	3.2+	10+	10+

Summary

Use `indexedDB` to store large sets of data that needs to be searchable.

Web Storage

What it is…

The Web Storage API provides objects for storing temporary (sessionStorage) and permanent (localStorage) data on the client's device.

Support

Chrome	Android	Safari	iOS	Firefox	IE
					
4+	2.1+	4+	3.2+	3.5+	8+

Session Storage

Stores data for current session and browser tab only.

```
sessionStorage
```

Local Storage

Provides a Storage object for an origin, that remains persistent even after restarting the browser.

```
localStorage
```

Testing Support

Vanilla javascript:

```
if(window.localStorage !== undefined){
  // window.localStorage is available!
}
```

Using Modernizr:

```
if(Modernizr.localstorage){
  // window.localStorage is available!
}
```

Syntax

```
// Setting
localStorage.setItem("foo", "Bar");
localStorage["foo"] = "Bar";
localStorage.foo = "Bar"

// Getting
localStorage.getItem("foo");
localStorage["foo"];
localStorage.foo;
```

Syntax

```
localStorage.setItem("foo", "Bar");
localStorage.getItem("foo");

// Removing an item
localStorage.removeItem("foo");

// Clearing storage
localStorage.clear();
```

Storing Data: numbers

```
localStorage.setItem("foo", 500);
> localStorage.getItem("foo");
> "500"

> ParseInt(localStorage.getItem("foo"), 10);
> 500
```

Storing Data: objects/arrays

```
var my_array = ["foo", "bar", "baz"];
localStorage.setItem("bar", JSON.stringify(my_array));
```

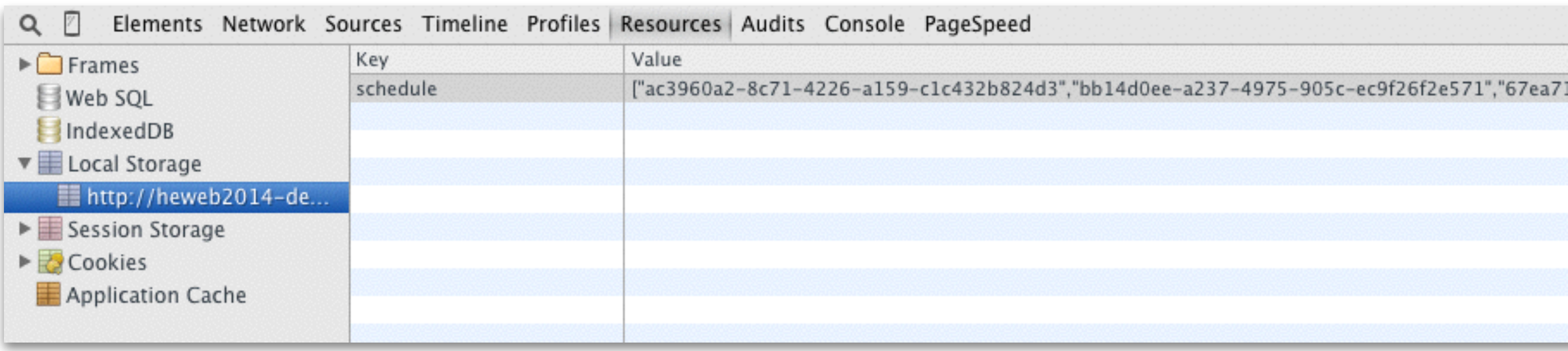
```
var my_stored_array = JSON.parse(localStorage.getItem("bar"));
```

Storage Limits

Chrome	Android	Safari	iOS	Firefox	IE
					
10MB	10MB	5MB	5MB	10MB	10MB

http://www.html5rocks.com/en/tutorials/offline/quota-research/

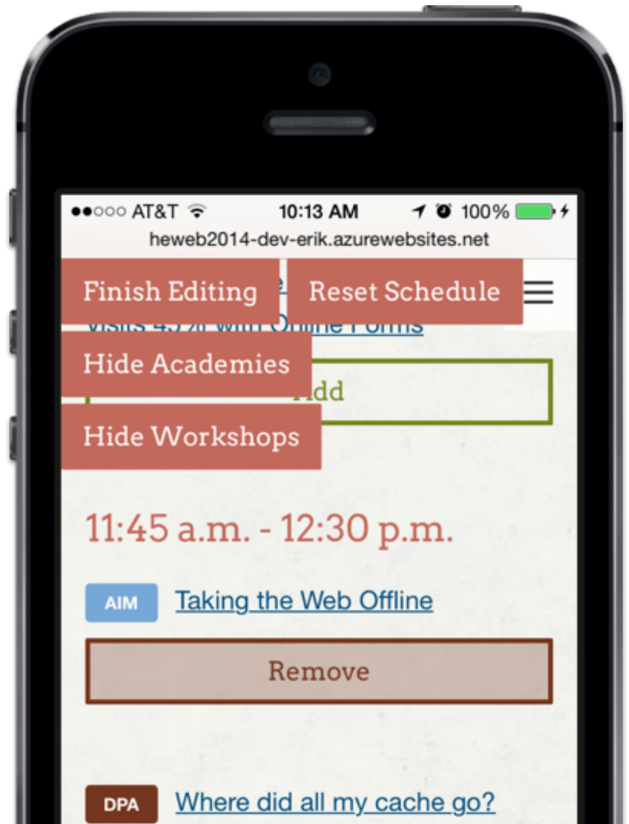
Chrome Inspector



2014.heweb.org

Editing the schedule

```
<li data-id="ac3960a2-8c71-4226-a159-c1c432b824d3">
  <a class="show-more" href="/long/url">Taking the Web Offline</a>
  <div class="data-more">
    <h4>Presenters</h4>
    <ul>
      <li>Erik Runyon - University of Notre Dame</li>
    </ul>
    <div class="session-abstract">Blah blah blah</div>
    <h4>Tags</h4>
    <ul>
      <li><a href="/schedule/tag/data%20and%20apis">data and APIs</a></li>
      <li><a href="/schedule/tag/responsive%20web%20design">responsive web design</a></li>
      <li><a href="/schedule/tag/front-end%20developers">front-end developers</a></li>
      <li><a href="/schedule/tag/advanced%20techniques">advanced techniques</a></li>
    </ul>
    <a class="btn" href="/long/url/">View Details</a>
  </div>
  <a class="btn-schtoggle btn btn-add" href="#">Add</a>
</li>
```



Adding/Removing

```
var schedule = JSON.parse(localStorage.getItem("schedule")) || [];

// As buttons are clicked
// sid is the pulled from the “data-id”
if($.inArray(sid, schedule) == -1){
  schedule.push(sid);
} else {
  var indexId = $.inArray(sid, schedule);
  schedule.splice(indexId, 1);
}

localStorage.setItem("schedule", JSON.stringify(schedule));
```

Workshops and Academies

```
foo.on("click", ".sch-academies", function(e){
  e.preventDefault();
  if(localStorage.getItem("hideAcademies") == "true"){
    localStorage.removeItem("hideAcademies");
    $(".section-academies").show();
    $(this).html("Hide Academies");
  } else {
    localStorage.setItem("hideAcademies", "true");
    $(".section-academies").hide();
    $(this).html("Show Academies");
  }
})
```

Workshops and Academies

```
foo.on("click", ".sch-academies", function(e){
  e.preventDefault();
  if(localStorage.getItem("hideAcademies") == "you_betcha"){
    localStorage.removeItem("hideAcademies");
    $(".section-academies").show();
    $(this).html("Hide Academies");
  } else {
    localStorage.setItem("hideAcademies", "you_betcha");
    $(".section-academies").hide();
    $(this).html("Show Academies");
  }
});
```

Items of note

- Clearing a browsers cache does **NOT** clear web storage
- There is no built-in way to expire web storage
- Data is scoped to the domain

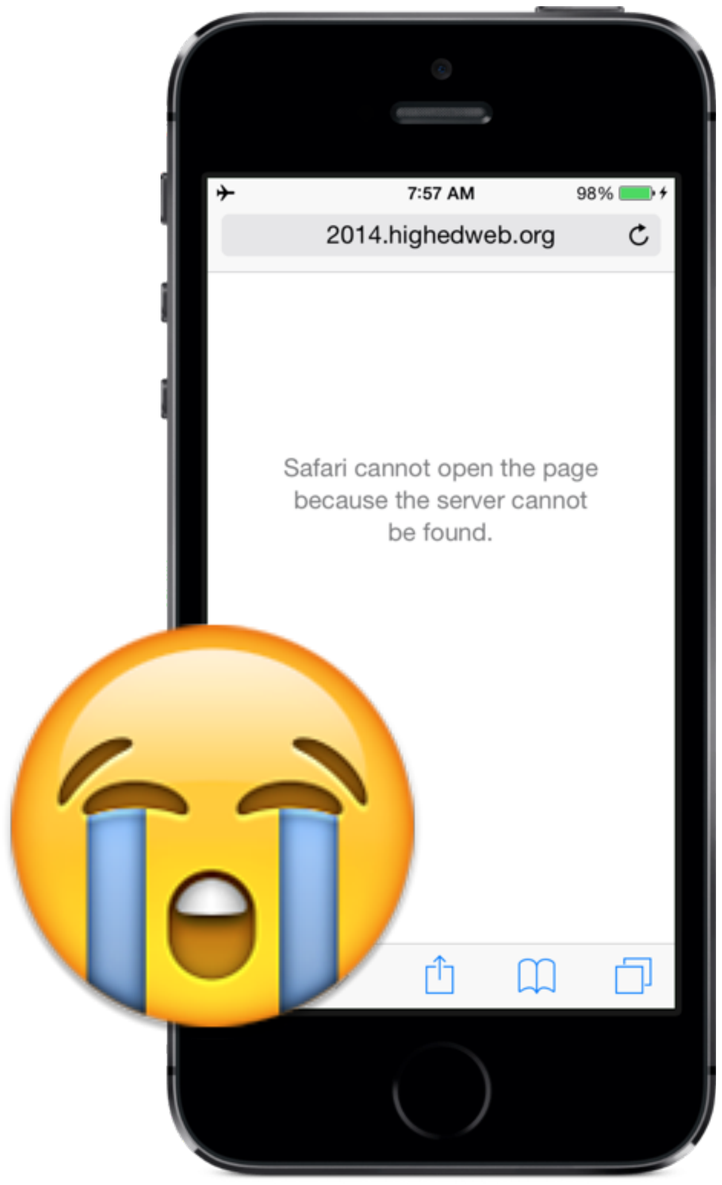
Summary

Use `localStorage` to quickly and easily store small amounts of data.

Appcache

What it is…

Application Cache provides a manifest which lists the files that are needed for the Web application to work offline and which causes the user's browser to keep a copy of the files for use offline.

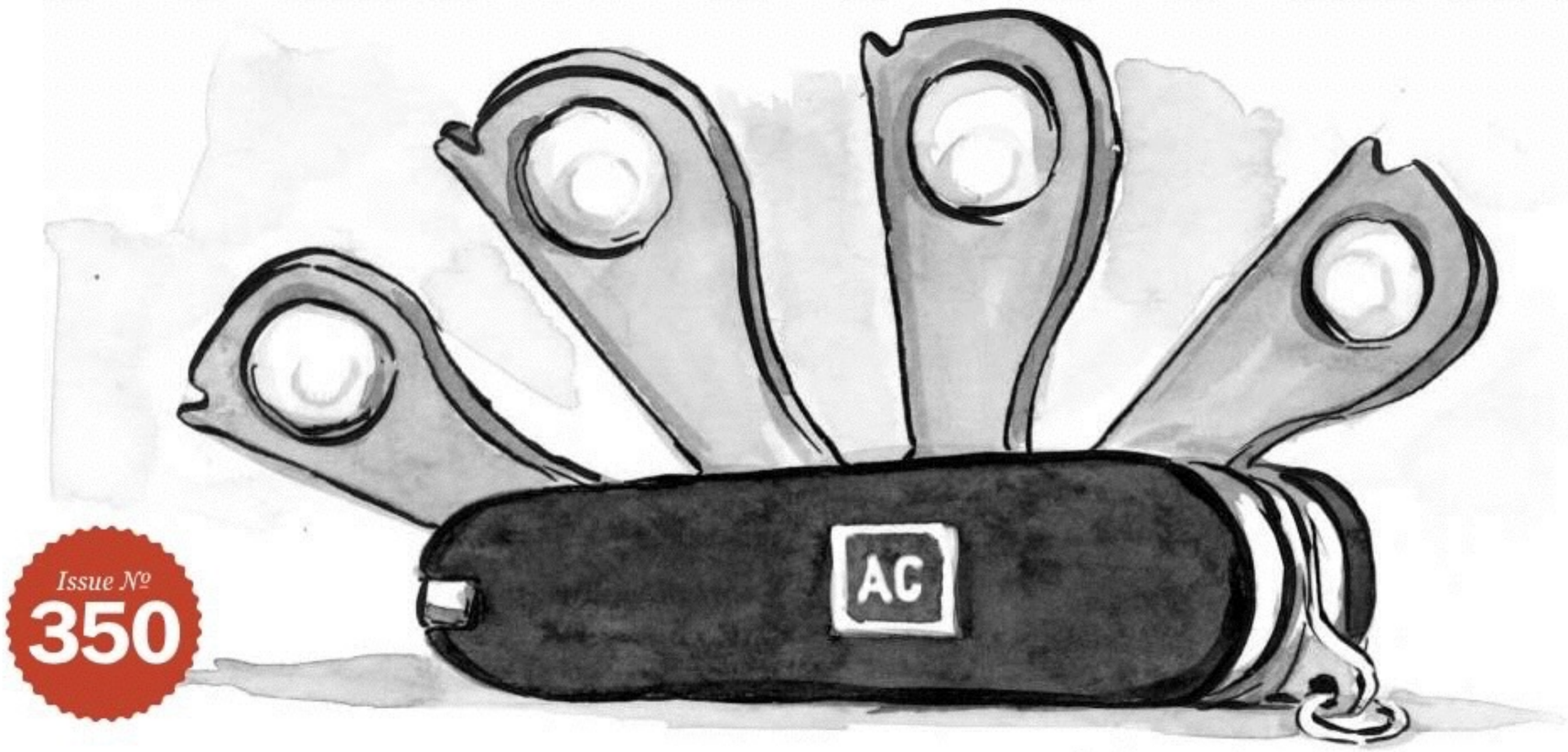

7:57 AM
98%
2014.highedweb.org
Safari cannot open the page because the server cannot be found.


9:10 AM
100%
2014.highedweb.org
HIGHEDWEB
ASSOCIATION
ANNUAL CONFERENCE 2014
OCT. 19-22
PORTLAND
OREGON
-22, 2014
annual conference of
tion Web Professionals
ated by and for all

Support

Chrome	Android	Safari	iOS	Firefox	IE
					
4+	2.1+	4+	3.2+	3.5+	10+

A LIST APART



Issue № 350

Application Cache is a Douchebag

by **JAKE ARCHIBALD** · *May 08, 2012*

Published in Application Development, HTML, JavaScript · 48 Comments

Good morning! Over in "castle Lanyrd" we recently launched our mobile site, which caches data on events you're attending for viewing offline. I've boiled the offline bits down to a simple demo and posted all the code on Github. But before we delve into the code, let

http://alistapart.com/article/application-cache-is-a-douchebag

Getting started…

mime-type

```
# Apache
AddType text/cache-manifest .appcache
```

```
# .NET
<mimeMap fileExtension=".appcache"
mimeType="text/cache-manifest" />
```

Prevent `appcache` Caching

```
# Apache
<IfModule mod_expires.c>
  ExpiresActive On
  ExpiresByType text/cache-manifest "access plus 0 seconds"
</IfModule>

# .NET
<location path="site.appcache">
  <system.webServer>
    <staticContent>
      <clientCache cacheControlMode="DisableCache" />
    </staticContent>
  </system.webServer>
</location>
```

Markup

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

```
<!doctype html>
<html lang="en" manifest="/example.appcache">
```

The manifest

```
CACHE MANIFEST
/foo/
stylesheet.css
images/logo.png
scripts/main.js
http://cdn.example.com/scripts/main.js
```

Ok, so it’s not quite THAT easy.

Notes on Domains

- Resources do NOT have to be on the same domain to be cached.
- Over SSL, all resources in the manifest must respect the same-origin policy (except in Chrome).

Basic Structure

```
CACHE MANIFEST
# v2014.10.21.0

CACHE:
/css/site.css

NETWORK:
*

FALLBACK:
/file.php /static.html
/images/ /images/offline.png
```

Required at the beginning of the file

List of explicit URLs to be stored locally

Resources only available while online

Instructions for requested files/paths that are not cached

The order of things

1. Browser visits site for the first time, site is downloaded
2. Manifest is read and all files are saved for offline use
3. Visitor returns to site and appcache serves page and assets from the cache (even if user is online)
4. Browser then checks for updated manifest
5. If update is found, appcache refreshes outdated files/assets
6. On next visit/refresh, the browser shows the most recent version

Items of note

- The referencing file will ALWAYS be cached
- The file referencing the manifest can't be NETWORK whitelisted
- The manifest file must be updated for changes to be sent
- If any files specified in the manifest cannot be found, the entire cache will be ignored
- Cached files are always served from appcache, even when online

Fonts

Include as few fonts as possible

Google Fonts

Referenced:
```
@import url(http://fonts.googleapis.com/css?family=Montserrat);
@import url(http://fonts.googleapis.com/css?family=Indie+Flower);
```

Downloaded:
```
http://fonts.gstatic.com/s/montserrat/v6/zhcz-_WihjSQC0oHJ9TCYAsYbbCjybiHxArTLjt7FRU.woff2
http://fonts.gstatic.com/s/indieflower/v7/10JVD_humAd5zP2yrFqw6hVuXpl7XtNjpLlhhhGlVqc.woff2
```

WOFF Support (86.6% U.S.A.)

Chrome	Android	Safari	iOS	Firefox	IE
5+	4.4+	5.1+	5.1+	3.6+	9+

http://caniuse.com/#feat=woff

The javascript side of appcache

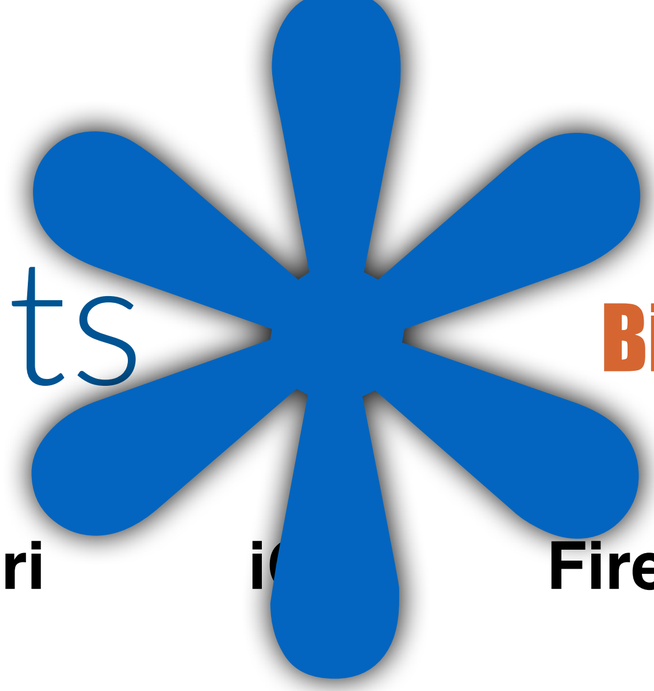
Javascript Events

updateready // Fired when the manifest resources have been newly re-downloaded
progress // Fired for each resource listed in the manifest as it is being fetched
checking // Checking for an update. Always the first event fired in the sequence
downloading // An update was found. The browser is fetching resources
cached // Fired after the first cache of the manifest
noupdate // Fired after the first download of the manifest
obsolete // This results in the application cache being deleted
error // The manifest returns 404 or 410, the download failed, or the manifest changed while the download was in progress

Forcing an update

```
// Check for updated version of appcache
window.addEventListener('load', function(e) {
  if(window.applicationCache) {
    var appCache = window.applicationCache;
    appCache.addEventListener('updateready', function(e) {
      if(appCache.status == appCache.UPDATEREADY) {
        if(confirm('A new version of this site is available. Load it?')) {
          window.location.reload();
        }
      }
    }, false);
  }
}, false);
```

Storage Limits



Big Frickin Asterisk

Chrome	Android	Safari	iO	Firefox	IE
Unlimited	20MB?	Unlimited	10MB	Unlimited	10MB

http://www.html5rocks.com/en/tutorials/offline/quota-research/

Chrome Storage Limits

Sharing the pool

Temporary storage is shared among all web apps running in the browser. The shared pool can be up to half of the of available disk space. Storage already used by apps is included in the calculation of the shared pool; that is to say, the calculation is based on `(available storage space + storage being used by apps) * .5`.

Each app can have up to 20% of the shared pool. As an example, if the total available disk space is 50 GB, the shared pool is 25 GB, and the app can have up to 5 GB. This is calculated from 20% of half of the available disk space.

Chrome Storage Limits

Running out of storage

Once the storage quota for the entire pool is exceeded, the entire data stored for the least recently used host gets deleted. The browser, however, will not expunge the data in LocalStorage and SessionStorage. For data stored in other offline APIs, the browser deletes the data in whole and not in part so that app data doesn't get corrupted in unexpected ways.

As each app is limited to a maximum of 20% of the storage pool, deletion is likely only if the user is actively running more than five offline apps that are each using the maximum storage.

However, available storage space can shrink as users add more files on their hard drives. When the available disk space gets tight (Remember, the shared pool only gets half of the *current* available disk space), the browser deletes all the data stored for the least recently used host.

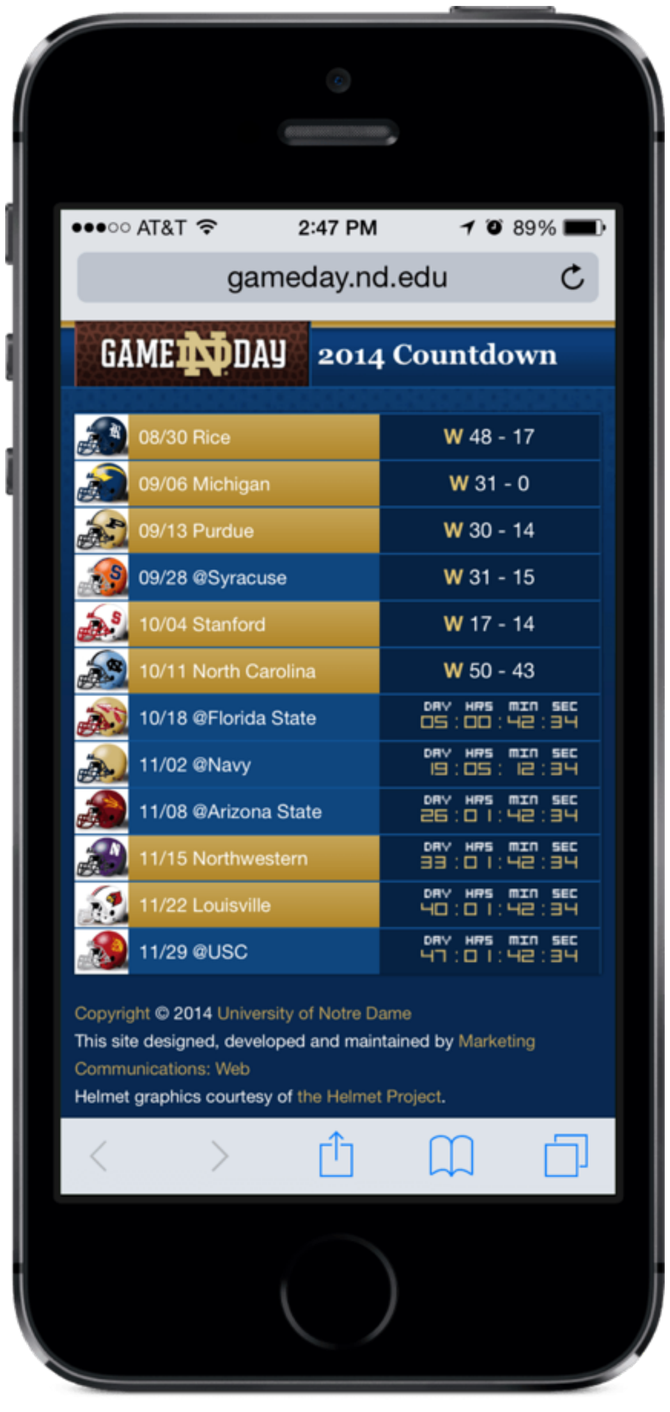
Storage Limits

Chrome	Android	Safari	iOS	Firefox	IE
It depends	It depends	It depends	It depends	It depends	It depends

http://www.html5rocks.com/en/tutorials/offline/quota-research/

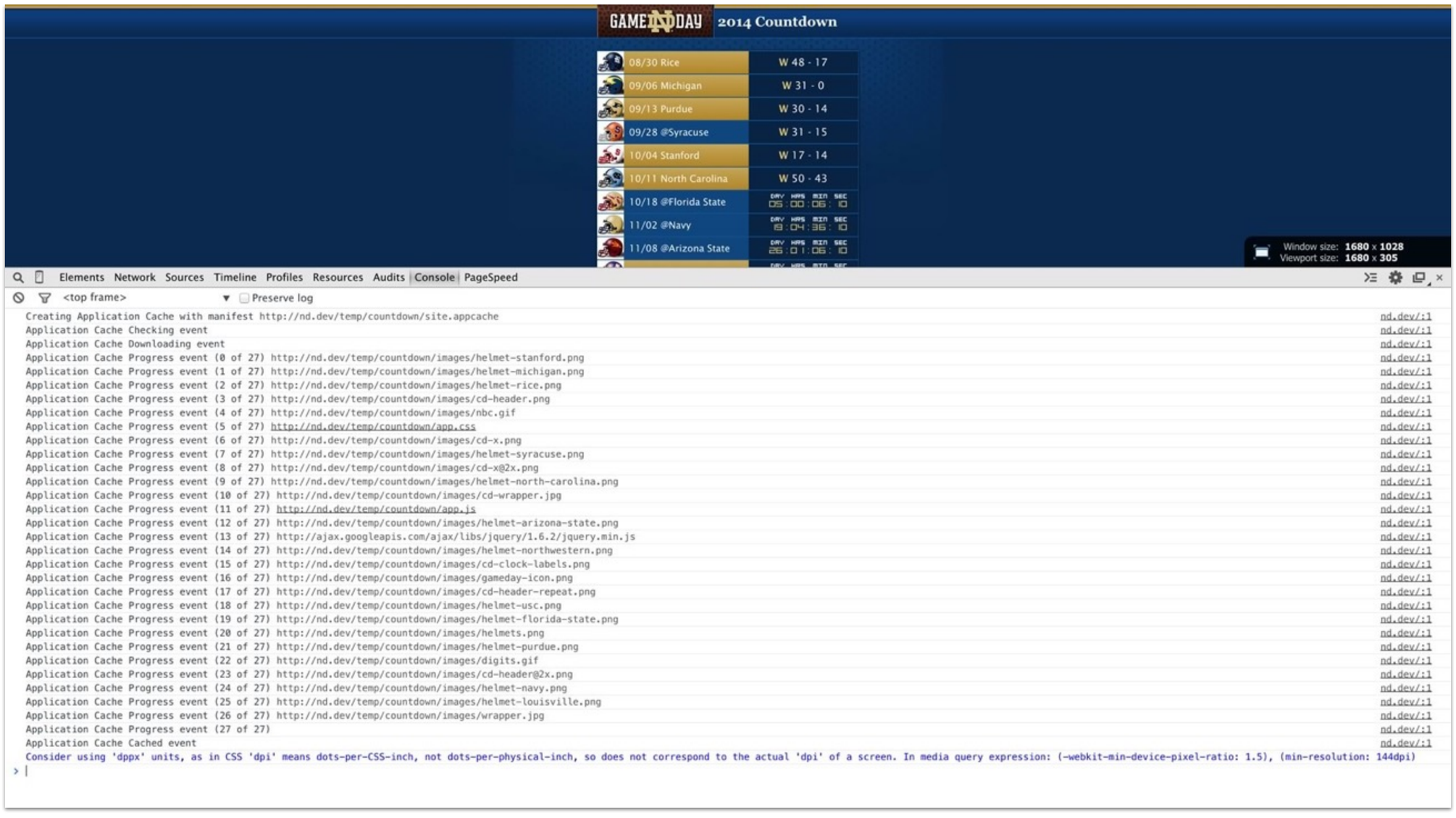
Example

gameday.nd.edu/countdown

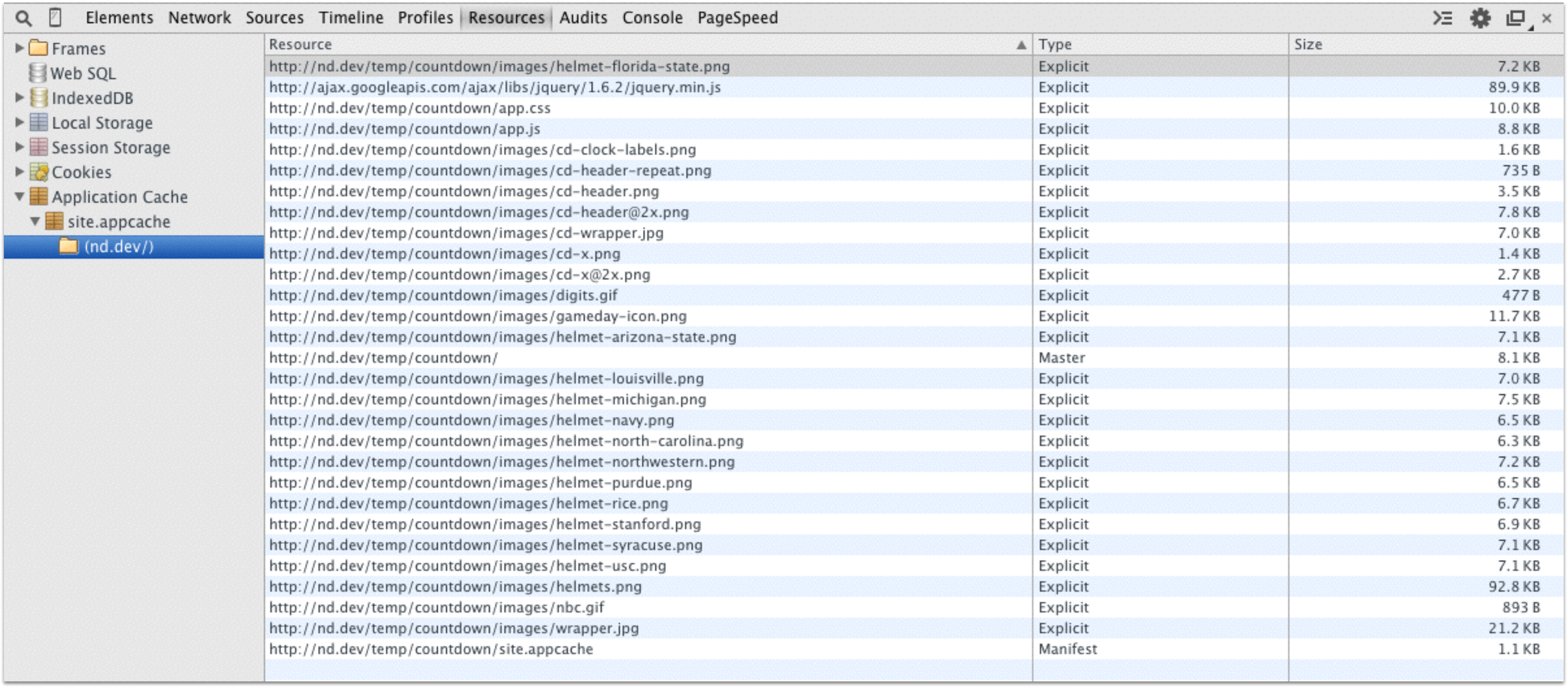


```
CACHE MANIFEST

# v2014.10.10.1

CACHE:
app.css
app.js
http://ajax.googleapis.com/ajax/libs/jquery/1.6.2/jquery.min.js

images/cd-clock-labels.png
images/cd-header-repeat.png
images/cd-header@2x.png
images/cd-wrapper.jpg
images/cd-x@2x.png
images/digits.gif
images/gameday-icon.png
images/helmet-arizona-state.png
images/helmet-florida-state.png
images/helmet-louisville.png
images/helmet-michigan.png
images/helmet-navy.png
images/helmet-north-carolina.png
images/helmet-northwestern.png
images/helmet-purdue.png
images/helmet-rice.png
images/helmet-stanford.png
images/helmet-syracuse.png
images/helmet-usc.png
images/nbc.gif
images/wrapper.jpg

NETWORK:
*
```

Download Progress



Chrome Inspector



Clearing appcache

chrome://appcache-internals/

Manifest: http://nd.dev/temp/countdown/site.appcache

Remove View Entries

- Size: 352 kB
- Creation Time: Monday, October 13, 2014 at 3:20:35 PM
- Last Update Time: Monday, October 13, 2014 at 3:20:35 PM
- Last Access Time: Monday, October 13, 2014 at 3:20:35 PM

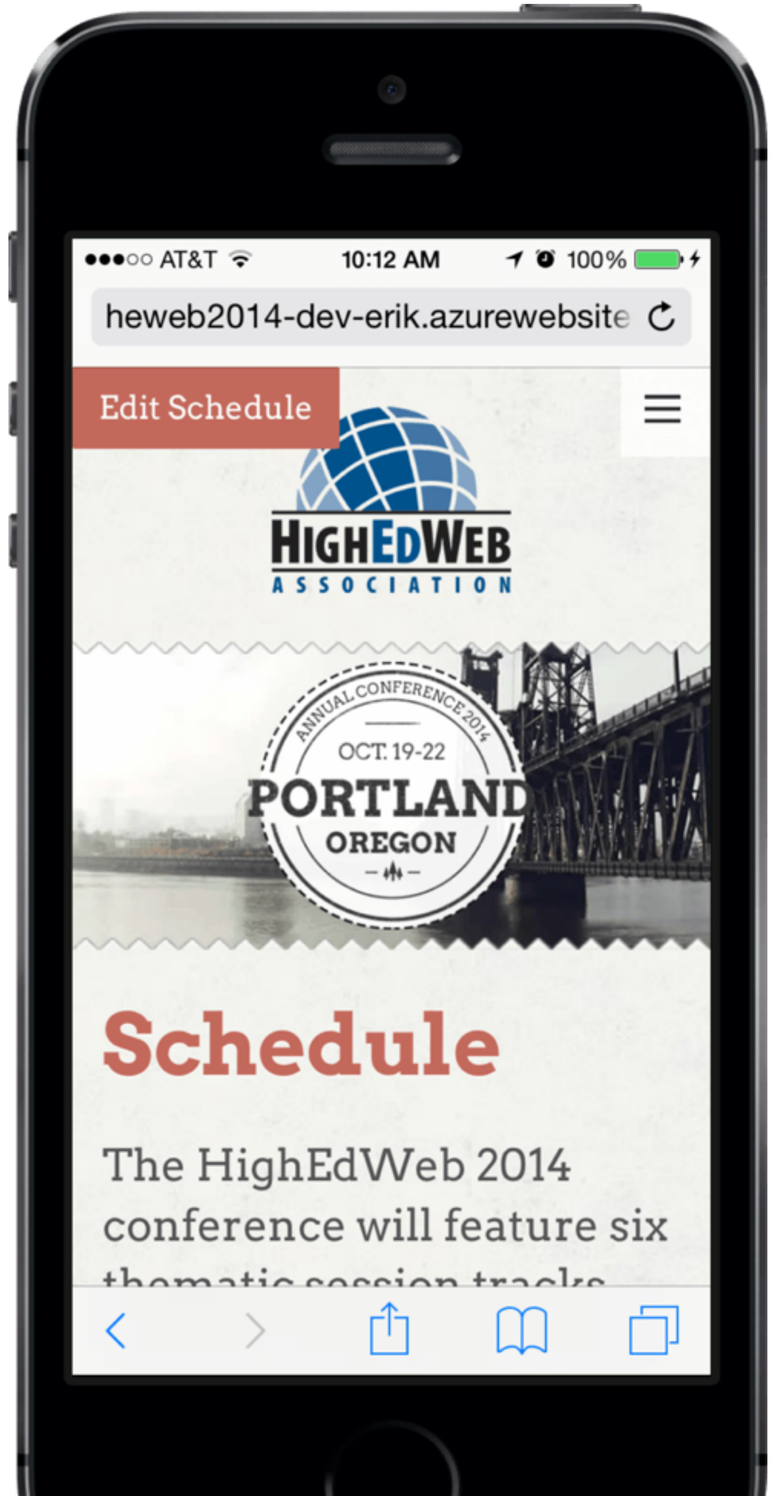
Example

2014.highedweb.org

Why?

Because conference wifi.

Instead of several “native” mobile apps, the conference committee decided to focus on an offline capable responsive website.



The pages

```
/
/about/
/attendees/
/schedule/
/events/
/venue/
/sponsors/
/gunn/
/hardwick/
```

The pages

```
/
/images/2014/gunn-home.jpg
/images/2014/hardwick-home.jpg
/about/
/images/2014/attendees.jpg
/attendees/
/images/2014/registration.jpg
/schedule/
/events/
/images/2014/afterdark.jpg
/images/2014/hackathon.jpg
```

The images

```
/sponsors/logos/acquia.png
/sponsors/logos/active-data.png
/sponsors/logos/barkley-rei.png
/sponsors/logos/beacon.png
/sponsors/logos/c2.png
/sponsors/logos/cactus.png
/sponsors/logos/campus-bird.png
/sponsors/logos/campusm.png
/sponsors/logos/campuspress.png
/sponsors/logos/expertfile.png
/sponsors/logos/form-assembly.png
```

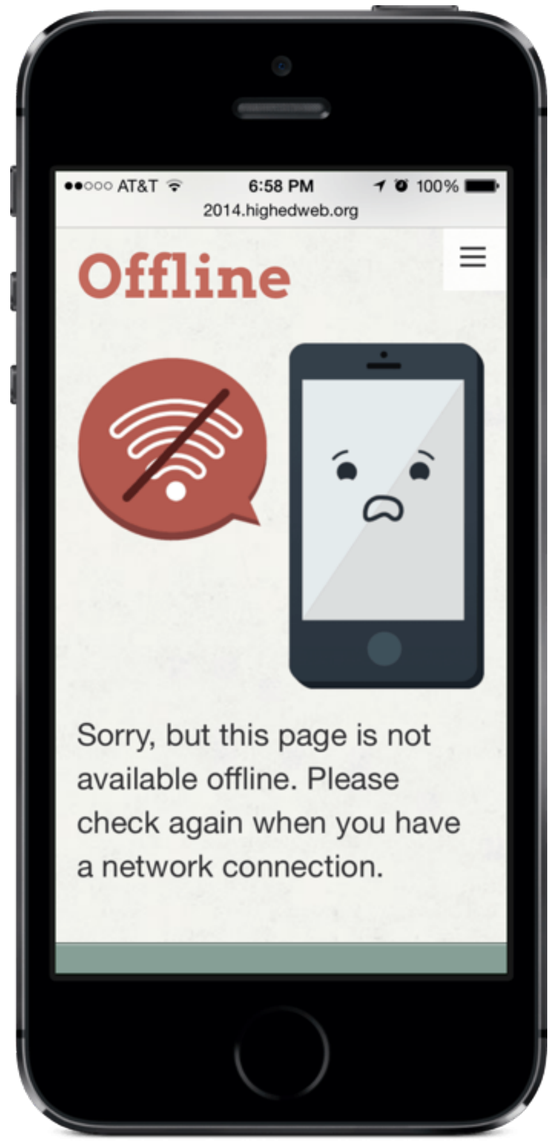
28 Images? Nope, sorry

The support staff

```
/styles/site.css
/styles/schedule.css

/styles/fonts/Arvo-Regular.woff
/styles/fonts/Arvo-Bold.woff

/scripts/modernizr.js
/scripts/jquery-2.1.0.min.js
/scripts/2014.js
```

Fallback pages

```
FALLBACK:
/page-with-a-form/ /offline/forms/
/ /offline/
```



Non-obtrusive updating

```
if(appCache.status == appCache.UPDATEREADY) {
  $('#site-reload').slideToggle();
}
```

IV. Wrapping Up

IndexedDB

For large sets of searchable data.

Web Storage

For small data sets (articles/settings) that don’t necessarily need to be passed to the server on every request.

Appcache

Store site/data for offline use.

Find a reason to play with new web technology.

Credits

Offline phone graphic: @tpacket
Browser logos: https://github.com/alrra/browser-logos

Photos:
Subway: https://www.flickr.com/photos/kenstein/2313111565
Notre Dame Stadium: Matt Cashore - photos.nd.edu
Cat: https://www.flickr.com/photos/77654185@N07/9029847786

thank you

Erik Runyon
@erunyon
weedygarden.net

bit.ly/hew2014