

HTML5 and CSS3 for Mobile Applications

Offline Working

Prof. Paul Krause, University of Surrey

What we are going to do

- Application cache
- Session Storage
- Local Storage

Application cache

AppCache API

- specify which files should be cached
- Supports:
 - offline browsing/working/playing
 - faster reloads
 - reduced server load
- Max of 5MB local storage on most mobile browsers

Just add a manifest file and you're ready to go...

```
1 <!DOCTYPE html>
2 <html lang="en-GB" manifest="offline.appcache">
3   <head>
4     <meta charset="utf-8" />
5     <title>Shopping List</title>
6     <script src="http://code.jquery.com/jquery-2.1.1.min.js"></script>
7     <script src="scripts/shopping.js"></script>
8     <link rel="stylesheet" type="text/css" href="stylesheets/normalize.css">
9     <link rel="stylesheet" type="text/css" href="stylesheets/shoppingInStyle.css">
10    <meta name="description" content="Simple shopping list app that demonstrates
11    off line working using local and session storage." />
12  </head>
13  <body>
14    <header>
15      <h1>Awsome Offline Shopping List App</h1>
16    </header>
```

<filename>.appcache

```
1  CACHE MANIFEST
2  # 18 June 2014 16:03
3
4  # explicitly cached entries
5  CACHE:
6  index.html
7  scripts/shopping.js
8  http://code.jquery.com/jquery-2.1.1.min.js
9  stylesheets/normalize.css
10 stylesheets/shoppingInStyle.css
11
12 # resources that require connectivity to function
13 NETWORK:
14 login.html
15
16 FALLBACK:
17 / /offline.html
```

Let's try it!

Updating the Appcache

- The app cache will only be refreshed if the manifest file is changed
- A common practice is to give the file a date stamp or version number
- Reading an edited manifest file will trigger a refresh
- The new files will not be used until the next time the application is started up

Session storage

Session storage

- Session storage data is:
 - confined to the browser window it was created in
 - accessible to any page from the same origin within that window
 - is deleted when the session ends
 - stored as key:value pairs

methods on the sessionStorage object

- `setItem(key, value)`
- `getItem(key)`
- `removeItem(key)`
- `clear()`
- `key(position)`
- `length`
- **N.B.** Hitting the hard drive on a mobile device is generally more performant (and energy efficient) than making an HTTP request

Saving the username to SessionStorage

```
1  $(function() {  
2    var userName = prompt("Please enter your name");  
3    sessionStorage.setItem('userName', userName);  
4  })  
5
```

Try me!

LocalStorage

Local Storage

- Local Storage data is:
 - accessible across all windows in the same browser
 - is accessible to any page from the same origin within the browser
 - persists after the browser (window) is closed
 - stored as key:value pairs

methods on the localStorage object

- `setItem(key, value)`
- `getItem(key)`
- `removeItem(key)`
- `clear()`
- `key(position)`
- `length`
- **N.B.** Hitting the hard drive on a mobile device is generally more performant (and energy efficient) than making an HTTP request

saving application data to localStorage

```
5    $('#submitItem').click(function(event) {  
6        var item = $('#item').val();  
7        $('#item').val("");  
8        localStorage.setItem('newItem', item);  
9    })
```

Try me!

What we have done

- Provided an overview of the support for offline working
- Shown how you can use the app cache to deliver an updatable application into any mobile device
- Covered record-based storage in the browser
 - Databases are also supported